

# STOCHASTIC ROUTING OPTIMIZED FOR AUTONOMOUS DRIVING

Inauguraldissertation  
zur Erlangung des akademischen Grades  
eines Doktors der Naturwissenschaften  
der Universität Mannheim

vorgelegt von  
M. Sc. Adam Samara  
aus München

Mannheim, 2021

Dekan: Dr. Bernd Lübcke, Universität Mannheim  
Referentin: Prof. Dr. Simone Göttlich, Universität Mannheim  
Korreferent: Prof. Dr. Michael Herty, RWTH Aachen University

Tag der mündlichen Prüfung: 19.07.2021

# Abstract

In this thesis, we propose a novel algorithm for stochastic routing optimized for autonomous vehicles. The key idea of stochastic routing is to include information on travel time reliability, rather than only estimating travel time itself. Travel time reliability is of major importance for travelers and transportation managers as it simplifies decision making and schedule planning. The concept of stochastic routing is then extended to fit the specific needs for an optimal autonomous drive. In near future, when vehicles enabled with fully autonomous driving become available, the autonomous driving features will only be possible on roads that fulfil certain criteria. Thus, when searching for an optimal route for one origin-destination-pair, we are not only interested in the travel time, but also on the route's properties concerning autonomous driving.

We estimate path travel time reliability by using empirical travel time data on segment-level. For that purpose we dive into the mathematical area of probability theory. First, we measure dependence between road segments. Then we use copulas for estimating travel time distribution on path-level by including the dependence between neighbouring road segments. In order to improve efficiency, which is needed for a real-world application, we use the following hybrid approach. We take convolution, which assumes independence, and extend it to the dependent case by integrating copulas, referred to as copula-based Dependent Discrete Convolution (DDC). Based on DDC we develop a methodology for stochastic routing.

We formulate a multicriteria optimization problem, in order to find a route optimized for an autonomous drive. Different approaches to obtain one optimal solution from the Pareto front are compared, and the best fitting one is selected. This framework is then combined with the stochastic routing methodology.





# Zusammenfassung

Diese Arbeit präsentiert einen neuartigen stochastischen Router, welcher für autonomes Fahren optimiert wurde. Ein stochastischer Router gibt Informationen zur Verlässlichkeit der Reisezeit an. Die Verlässlichkeit von Reisezeiten vereinfacht sowohl das Treffen von reisebedingten Entscheidungen als auch logistische Zeitplanungen und ist somit von hoher Relevanz für Reisende und Transportmanager. Das Konzept des stochastischen Routers wird erweitert, um auf die spezifischen Anforderungen für autonomes Fahren einzugehen. In naher Zukunft wird vollautonomes Fahren voraussichtlich nur auf einzelnen Streckenabschnitten möglich sein, welche bestimmte Kriterien erfüllen. Bei der Suche nach einer optimalen Route für eine Start-Ziel-Beziehung werden daher zusätzlich zur Reisezeit auch weitere Kriterien bezüglich autonomen Fahrens relevant sein.

Wir schätzen die Verlässlichkeit der Reisezeit einer Route mithilfe von empirischen Reisezeitdaten von Streckensegmenten. Zu diesem Zweck begeben wir uns in den mathematischen Bereich der Wahrscheinlichkeitstheorie. Zunächst analysieren wir die Abhängigkeiten zwischen den Streckensegmenten. Dann benutzen wir Copulas um Reisezeitverteilungen für eine gesamte Route ausgehend von den Daten der einzelnen Segmente zu schätzen. Anschließend präsentieren wir einen Hybridansatz, um die Effizienz der Methodik zu steigern. Wir erweitern die Faltung, welcher die Annahme der Unabhängigkeit zwischen Operanden zugrunde liegt, zu einem Operator, der Abhängigkeiten berücksichtigen kann. Dafür integrieren wir Copulas in die Formulierung der Faltung, welche dann als „Dependent Discrete Convolution“ (DDC) bezeichnet wird. Basierend auf der DDC entwickeln wir eine Methodik für einen stochastischen Router.

Wir formulieren ein multikriterielles Optimierungsproblem, um die optimale Route für eine autonome Fahrt zu finden. Verschiedene Ansätze werden miteinander verglichen und der am besten geeignete Ansatz wird gewählt. Dieses Modell wird dann mit der Methodik des stochastischen Routers kombiniert.



# Danksagung

Es ist mir ein Anliegen, mich bei den Menschen zu bedanken, die mir diese Dissertation ermöglicht haben. Anfangen möchte ich bei Prof. Dr. Simone Göttlich, die mir an ihrem Lehrstuhl die Möglichkeit zur Promotion gab. Ich möchte mich bei Ihr für die hervorragende Betreuung und die angenehme Zusammenarbeit bedanken, durch die ich sehr viel lernen konnte.

Mein Dank geht auch an Dr. Felix Rempe, der mich auf der Unternehmensseite betreut hat. Danke für die zahlreichen Austausche, die stetige Hilfsbereitschaft, und die angenehme Zusammenarbeit.

Außerdem möchte ich mich bei Prof. Dr. Michael Herty bedanken, der sich als Korreferent bereit erklärt hat, das Zweitgutachten meiner Arbeit zu erstellen.

Ich möchte auch meinen Eltern danken, die mich auf meinem gesamten Lebensweg begleitet und unterstützt haben.



# Acronyms

<b>AD</b>	Autonomous Driving
<b>AVI</b>	Automated Vehicle Identification
<b>CDF</b>	Cumulative Distribution Function
<b>CVM</b>	Cramer-von-Mises statistic
<b>DDC</b>	Dependent Discrete Convolution
<b>DM</b>	Decision Maker
<b>EM</b>	Expectation-Maximization
<b>FCD</b>	Floating Car Data
<b>FFT</b>	Fast Fourier Transformation
<b>GMM</b>	Gaussian Mixture Model
<b>GNSS</b>	Global Navigation Satellite System
<b>KS</b>	Kolmogorov-Smirnov statistic
<b>PDF</b>	Probability Density Function
<b>SPP</b>	Shortest Path Problem
<b>SSP</b>	Stochastic Shortest Path Problem
<b>WPM</b>	Weighted Product Model



# Contents

<b>Introduction</b>	<b>1</b>
<b>1 Probability Theory</b>	<b>5</b>
1.1 Preliminaries . . . . .	5
1.2 Random Variables and Distributions . . . . .	7
1.3 Dependence . . . . .	10
1.3.1 Copulas . . . . .	10
1.3.2 Correlation . . . . .	13
1.3.3 Dependent Discrete Convolution . . . . .	16
<b>2 Shortest Path Problem and Data Sampling</b>	<b>19</b>
2.1 Shortest Path Problem . . . . .	19
2.1.1 Problem Formulation . . . . .	19
2.1.2 Dijkstra's Algorithm . . . . .	20
2.1.3 Yen's $k$ -shortest Path Algorithm . . . . .	22
2.2 Data . . . . .	24
2.2.1 Sampling and Collection . . . . .	24
2.2.2 Preprocessing . . . . .	25
2.2.3 Data Set . . . . .	25
<b>3 Modelling travel time reliability</b>	<b>27</b>
3.1 Related Work . . . . .	27
3.2 Copula Model . . . . .	29
3.2.1 Methodology . . . . .	30
3.2.2 Goodness of Fit Tests . . . . .	31
3.2.3 Case Study: Urban Arterial . . . . .	32
3.2.4 Case Study: Freeway Arterial . . . . .	42
3.3 DDC Model . . . . .	50
3.3.1 Methodology . . . . .	51
3.3.2 Comparison to Copula Model . . . . .	52

<b>4</b>	<b>Stochastic Routing</b>	<b>57</b>
4.1	Related Work . . . . .	57
4.2	Preliminaries . . . . .	59
4.3	Methodology . . . . .	60
4.3.1	Calculating Alternate Routes . . . . .	60
4.3.2	Generalized DDC Model . . . . .	61
4.3.3	Stochastic Router Formulation . . . . .	63
4.4	Case Study . . . . .	63
<b>5</b>	<b>Routing Optimized for Autonomous Driving</b>	<b>69</b>
5.1	Methodology . . . . .	70
5.1.1	A Subjective Approach . . . . .	70
5.1.2	Multiobjective Optimization Problem Formulation . . . . .	72
5.2	Case Studies . . . . .	75
5.2.1	Munich – Frankfurt . . . . .	75
5.2.2	Stuttgart – Cologne . . . . .	77
5.2.3	Frankfurt – Dresden . . . . .	77
5.2.4	Summary . . . . .	79
5.3	Combination with Stochastic Routing . . . . .	83
<b>6</b>	<b>Conclusion and Outlook</b>	<b>85</b>



# Introduction

Let us begin with the following scenario. We need to travel from an origin to a destination. For that matter, we want to obtain the route with the minimal travel time. In addition, we need to be at our destination at a sharp deadline. A state-of-the-art routing device can calculate the shortest route and gives us the estimated travel time. However, we do not know how reliable that estimated travel time is. As we are bound to our deadline, we need to plan some buffer in case of a possible variance in travel time. But how much buffer should we plan? Should we rather plan a small buffer and risk that we arrive late, or should we plan a large buffer and risk losing a lot of time waiting at the destination? Both alternatives are not optimal. But what if we had a routing device that could give us a route with a reliable travel time with a 90%, 95%, or even a 100% probability of arriving on time? Then we could avoid this dilemma altogether, as we would not need to plan any buffer. In this thesis such a routing algorithm will be developed.

In our scenario reliability of travel time estimates is important for travelers. But it is also crucial for transport managers and logistics planners as it provides information for decision making and planning schedules. Furthermore, travel time reliability is one of the key indicators for evaluating the performance of transportation systems. In [40] it is shown that measures on travel time reliability can improve regional transportation planning and operations.

Providing travel time reliability in route guidance systems is the key idea of Stochastic Shortest Path Problem (SSP). Such an application is also referred to as *stochastic router*. While the traditional shortest path problem is defined in a static environment, the SSP problem considers also the uncertainties of traffic conditions. These uncertainties can originate, for example, from demand fluctuations, weather conditions, and accidents. Providing a solution in the SSP problem enables to include travel time reliability into route guidance systems.

Another scope of this thesis is to optimize the stochastic router for Autonomous Driving (AD). Before we elucidate why we need such an approach, we briefly discuss AD. The development of AD attracts extensive attention from both academy and industry. Based on the degree of transition from assisted driving towards fully autonomous vehicles, i.e. vehicles that do not need any human interaction, AD is divided into different Levels by SAE International [32]. Level 0 contains basic assisted driving features, e.g. automatic emergency braking, blind spot warning, and lane departure warning, whereas Level 5 describes full AD possible everywhere in all conditions. From Level 3, certain road sections can be driven completely autonomously. The property, that enables AD on a road, is called road clearance. In order to determine if road clearance can be activated for a particular road sections, several steps need to be performed. First an evaluation based on on-board sensor information is conducted. If the surrounding is accurately reconstructed and defined as safe, then the vehicle is allowed to drive autonomously. In addition, relevant information on the actual road segment is transferred from the vehicle to a central server, enabling to extend the limited horizon of the individual vehicle. Then, the vehicle requests the status of the road clearance from this server on a regular basis. The activation of road clearance depends on weather conditions, incidents, and on road conditions, such as missing lane marks or badly maintained roads. Thus, activation and deactivation of road clearance is a highly dynamic process. In case the road clearance is deactivated due to one of these reasons, the driver has to take over the control over the vehicle.

So why do we need to optimize route guidance systems for AD? At the time of writing this thesis, vehicles capable of Level 3 AD are announced by several automotive manufacturers to be released onto the market soon. When purchasing such a vehicle, we can assume that customers want to drive autonomously as much as possible. Thus, when providing route guidance, not only travel time becomes a criteria, but also which route is best suited for AD, which is highly dependent on the dynamics of road clearance.

The contributions of this thesis can be summarized as follows:

- Methodologies are developed for estimating travel time reliability.
- Based on these methodologies a stochastic router is designed.
- A routing framework optimized for autonomous driving is developed.

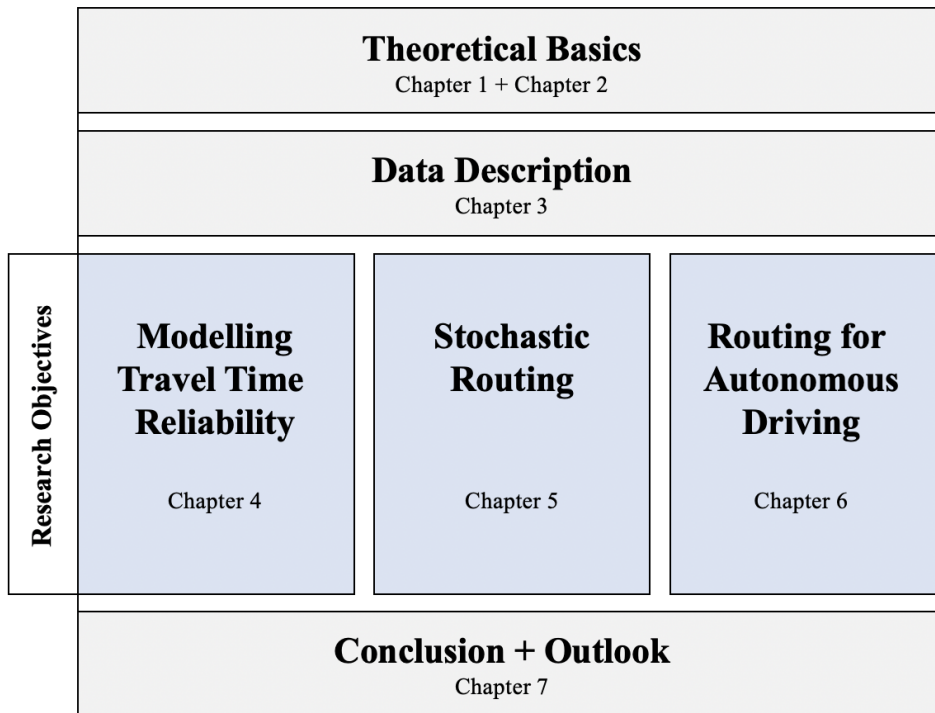


FIGURE 1: Structure of this thesis.

- An integrated approach for a stochastic router optimized for autonomous driving is proposed.

## Structure of this thesis

The outline of this thesis is illustrated in Figure 1. First we set the stage for our research objectives. We start by giving an overview of the underlying theoretical basics of this thesis. The mathematical area of probability theory is introduced to cover the preliminaries for the stochastic aspects of stochastic routing. There we will discuss random variables, distributions and the concept of dependence. As dependence between random variables plays a crucial role throughout this thesis, the focus is set on this topic especially. The next chapter covers the theoretical requirements of vehicle routing. We will present the shortest path problem as well as the  $k$ -shortest path problem and its solutions using Dijkstra's Algorithm and Yen's Algorithm, respectively.

Then we present the floating car data, which was used for this work in Chapter 3. We will elucidate how the raw data is sampled and collected from probe vehicles, before it is preprocessed in a format which contains the information we need.

In the next chapter, we begin with the research objective. First we discuss travel time reliability and give an overview of the state-of-the-art approaches for its modelling. Then we present the methodologies that we have developed together with case studies for evaluation which have been published as

- A. Samara, F. Rempe, U. Fastenrath and S. Göttlich, "Assessing the probability of arriving on time using historical travel time data in a road network", 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 2019, pp. 1343-1348.
- A. Samara, F. Rempe and S. Göttlich, "Modelling arterial travel time distribution using copulas", 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 2020, pp. 1-6.

In Chapter 5 we will use the methodologies developed in the previous chapter to design a stochastic router, after giving an overview of the state-of-the-art approaches. A case study is conducted in order to demonstrate the use case for a real world application. Results have been published as

- A. Samara, F. Rempe, S. Göttlich. "A novel approach for vehicle travel time distribution: copula-based dependent discrete convolution", *Transportation Letters*, 2021, pp. 1-12, DOI: 10.1080/19427867.2021.1941707.

Chapter 6 is devoted to routing optimized for autonomous vehicles. We will present two distinct methodologies, where the first one is based on subjective perception and the second one on multiobjective optimization. Case studies are conducted in order to compare both methods. Results have been published as

- A. Samara, F. Rempe and S. Göttlich, "Vehicle Routing Optimized for Autonomous Driving", 2021 32nd IEEE Intelligent Vehicles Symposium (IV), Nagoya, Japan, 2021.

Finally, we design a stochastic router optimized for autonomous vehicles by integrating the methodologies we have developed.

In the last chapter we give a conclusion as well as an outlook for potential future work.

# Chapter 1

## Probability Theory

In this chapter we set the theoretical basis for this thesis. In order to model stochastic processes such as travel time estimation, we employ tools from probability theory. We give an overview of probability spaces leading to the definition of random variables and their distribution functions. One major contribution of this work is the inclusion of dependence into the modelling of travel time distributions and, thus, into the development of the stochastic routing application. Therefore, the focus here is set on modelling dependence between random variables.

The definitions and theorems in this chapter are based on the following literature, if not stated otherwise. For the discussion of the theoretical preliminaries and random variables we follow the books "Probability: Theory and Examples" by Rick Durrett [20], "Measure Theory" by Donald Cohn [15] and "An Introduction to Measure Theory" by Terence Tao [56]. The section on dependence follows the books "An Introduction to Copulas" by Nelson Roger [3] and "Correlation and Dependence in Risk Management: Properties and Pitfalls" by Paul Embrechts et al. [21].

### 1.1 Preliminaries

Here we set the stage for our methodology. For that purpose, we introduce probability spaces.

**Definition 1.1.** A *probability space* is a triple  $(\Omega, \mathcal{F}, P)$  where  $\Omega$  is a set of outcomes,  $\mathcal{F}$  is a set of events, and  $P : \mathcal{F} \rightarrow [0, 1]$  is a function assigning probabilities to events.  $\mathcal{F}$  is assumed to be a  $\sigma$ -algebra, i.e. a nonempty collection of subsets of  $\Omega$  that satisfy

- (i) if  $A \in \mathcal{F}$  then  $A^c \in \mathcal{F}$ , where  $A^c$  is the complement of  $A$ , and
- (ii) if  $A_i \in \mathcal{F}$  is a countable sequence of sets then  $\cup_i A_i \in \mathcal{F}$ .

We refer to *countable* as finite or countably infinite. Because  $\cap_i A_i$ , a  $\sigma$ -algebra is therefore closed under countable intersections.

When removing  $P$  from the probability space, the remaining set  $(\Omega, \mathcal{F})$  is denoted as *measurable space*.

**Definition 1.2.** A *measure* is a nonnegative countably additive set function, i.e. a function  $\mu : \mathcal{F} \rightarrow \mathbb{R}$  with

- (i)  $\mu(A) \geq \mu(\emptyset) = 0$  for all  $A \in \mathcal{F}$ , and
- (ii) if  $A_i \in \mathcal{F}$  is a countable sequence of disjoint sets, then  $\mu(\cup_i A_i) = \sum_i \mu(A_i)$ .

If  $\mu(\Omega) = 1$ , we refer to  $\mu$  as a *probability measure*.

**Theorem 1.1.** Let  $\mu$  be a measure on  $(\Omega, \mathcal{F})$ . Then  $\mu$  has the following properties:

- (i) *monotonicity:* If  $A \subset B$  then  $\mu(A) \leq \mu(B)$ .
- (ii) *subadditivity:* If  $A \subset \cup_{m=1}^{\infty} A_m$  then  $\mu(A) \leq \sum_{m=1}^{\infty} \mu(A_m)$ .
- (iii) *continuity from below:* If  $A_1 \uparrow A$ , i.e.  $A_1 \subset A_2 \subset \dots \subset A_n$ , and  $\cup_i A_i = A$  then  $\mu(A_1) \subset \mu(A_2) \subset \dots \subset \mu(A_n)$ .
- (iv) *continuity from above:* If  $A_1 \downarrow A$ , i.e.  $A_1 \supset A_2 \supset \dots \supset A_n$ , and  $\cap_i A_i = A$ , with  $\mu(A_1) < \infty$ , then  $\mu(A_1) \downarrow \mu(A)$ .

*Proof.* (i) Let  $B - A = B \cap A^c$  be the difference of the two sets. We use  $+$  to denote disjoint union,  $B = A + (B - A)$  so

$$\mu(B) = \mu(A) + \mu(B - A) \geq \mu(A).$$

(ii) Let  $A' = A_n \cap A$ ,  $B_1 = A'_1$  and for  $n > 1$ ,  $B_n = A'_n - \cup_{m=1}^{n-1} A'_m$ . The  $B_n$  are disjoint and have union  $A$ , hence, when we use (ii) of Definition 1.2,  $B_m \subset A_m$ , and (i) of this theorem, we have

$$\mu(A) = \sum_{m=1}^{\infty} \mu(B_m) \leq \sum_{m=1}^{\infty} \mu(A_m).$$

(iii) Let  $B_n = A_n - A_{n-1}$ . Then the  $B_n$  are disjoint and have  $\cup_{m=1}^{\infty} B_m = A$ ,  $\cup_{m=1}^n B_m = A_n$  so

$$\mu(A) = \sum_{m=1}^{\infty} \mu(B_m) = \lim_{n \rightarrow \infty} \sum_{m=1}^n \mu(B_m) = \lim_{n \rightarrow \infty} \mu(A_n).$$

(iv)  $A_1 - A_n \uparrow A_1 - A$  so from (iii) it follows that  $\mu(A_1 - A_n) \uparrow \mu(A_1 - A)$ . Since  $A_1 \supset A$  we have  $\mu(A_1 - A) = \mu(A_1) - \mu(A)$  and, thus,  $\mu(A_n) \downarrow \mu(A)$ .  $\square$

The following two definitions are included, as we will encounter these terminologies in the next section.

**Definition 1.3.** Let  $\mathbb{R}^n$  be the set of vectors  $(x_1, \dots, x_n)$  of real numbers. Then,  $\mathcal{R}^n$  are the *Borel sets* defined as the smallest  $\sigma$ -algebra containing the open sets.

**Definition 1.4.** A set  $E \subset \mathbb{R}^n$  is referred to as *Lebesgue measurable* if, for every  $\epsilon > 0$ , there exists an open set  $U \subset \mathbb{R}^n$  containing  $E$  such that  $m^*(U \setminus E) \leq \epsilon$ . If  $E$  is Lebesgue measurable, we call  $m(E) := m^*(E)$  the *Lebesgue measure* of  $E$ .

## 1.2 Random Variables and Distributions

Now that we are familiar with probability spaces, we can define random variables on them. The representation of arterial travel times as random variables builds the core of the methodology presented in this thesis.

**Definition 1.5.** A real valued function  $X$  defined on  $\Omega$  is referred to as *random variable* if for every Borel set  $B \in \mathbb{R}$  we have  $X^{-1}(B) = \{\omega : X(\omega) \in B\} \in \mathcal{F}$ .

**Definition 1.6.** If  $X$  is a random variable, then  $X$  induces a probability measure on  $\mathbb{R}$  referred to as its *distribution* by setting  $\mu(A) = P(X \in A) = P(X^{-1}(A))$  for Borel sets  $A$ .

**Definition 1.7.** The distribution of a random variable  $X$  is described by giving its *distribution function*,  $F(x) = P(X \leq x)$ . When  $X$  has distribution function  $F$ , we denote  $X \sim F$ .

**Theorem 1.2.** Any distribution function  $F$  has the following properties:

- (i)  $F$  is nondecreasing.
- (ii)  $\lim_{x \rightarrow \infty} F(x) = 1, \lim_{x \rightarrow -\infty} F(x) = 0$ .
- (iii)  $F$  is right continuous, i.e.  $\lim_{y \downarrow x} F(y) = F(x)$ .
- (iv) If  $F(x-) = \lim_{y \uparrow x} F(y)$  then  $F(x-) = P(X < x)$ .
- (v)  $P(X = x) = F(x) - F(x-)$ .

*Proof.* (i) If  $x \leq y$ , then  $\{X \leq x\} \subset \{X \leq y\}$ , and then by using (i) of Theorem 1.1 we can conclude that  $P(X \leq x) \leq P(X \leq y)$ .

(ii) If  $x \uparrow \infty$ , then  $\{X \leq x\} \uparrow \Omega$ , and if  $x \downarrow -\infty$ , then  $\{X \leq x\} \downarrow \emptyset$ . Then we can use (iii) and (iv) of Theorem 1.1.

(iii) If  $y \downarrow x$ , then  $\{X \leq y\} \downarrow \{X \leq x\}$ .

(iv) If  $y \uparrow x$ , then  $\{X \leq y\} \uparrow \{X \leq x\}$ .

(v) Note that  $P(X = x) = P(X \leq x) - P(X < x)$  and use (iii) and (iv).  $\square$

**Theorem 1.3.** *If  $F$  satisfies (i), (ii), and (iii) in Theorem 1.2, then it is the distribution function of some random variable.*

*Proof.* Let  $\mathcal{F}$  be the Borel sets,  $P$  a Lebesgue measure and  $\Omega = (0, 1)$ . If  $\omega \in (0, 1)$ , let

$$X(\omega) = \sup\{y : F(y) < \omega\}.$$

If  $\omega \leq F(x)$  then  $X(\omega) \leq x$ , since  $x \notin \{y : F(y) < \omega\}$ . If  $\omega > F(x)$ , then since  $F$  is right continuous, there is an  $\epsilon > 0$  so that  $F(x + \epsilon) < \omega$  and  $X(\omega) \geq x + \epsilon > x$ . Therefore

$$\{\omega : X(\omega) \leq x\} = \{\omega : \omega \leq F(x)\}$$

and the desired result follows since  $P(\omega : \omega \leq F(x)) = F(x)$ .  $\square$

**Definition 1.8.** If  $X$  and  $Y$  induce the same distribution  $\mu$  on  $(\mathbb{R}, \mathcal{R})$  we say that  $X$  and  $Y$  are *equal in distribution*. This holds if and only if  $X$  and  $Y$  induce the same distribution function, i.e.  $P(X \leq x) = P(Y \leq x)$ .

**Definition 1.9.** When the distribution function  $F(x) = P(X \leq x)$  has the form

$$F(x) = \int_{-\infty}^x f(y)dy,$$

we say that  $X$  has *density function*  $f$ , satisfying  $f(x) \geq 0$  and  $\int f(x)dx = 1$ .

**Definition 1.10.** For  $X \geq 0$  the *expected value* is obtained by

$$E[X] = \int XdP.$$

$E[X]$  is called the *mean* of  $X$  and is denoted by  $\mu$ .

One important example of distributions which we will encounter during this thesis is the *normal distribution*, also referred to as *Gaussian distribution*

$$\phi(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \quad (1.1)$$



where  $\sigma = \sqrt{E[X^2] - (E[X])^2}$  is called the *standard deviation* of  $X$ .

In the following we study the relationship between random variables. Thus, we introduce the concept of independence.

**Definition 1.11.** Two events  $A$  and  $B$  are *independent* if  $P(A \cap B) = P(A)P(B)$ . Subsequently, two random variables  $X$  and  $Y$  are independent if for all  $C, D \in \mathcal{R}$ ,  $P(X \in C, Y \in D) = P(X \in C)P(Y \in D)$ .

**Theorem 1.4.** In order for  $X_1, \dots, X_n$  to be independent, it is sufficient that for all  $x_1, \dots, x_n \in \mathbb{R}$

$$P(X_1 \leq x_1, \dots, X_n \leq x_n) = \prod_{i=1}^n P(X_i \leq x_i). \quad (1.2)$$

**Theorem 1.5.** If  $X$  and  $Y$  with distribution functions  $F(x) = P(X \leq x)$  and  $G(y) = P(Y \leq y)$  are independent, then

$$\begin{aligned} P(X + Y \leq z) &= F(x) * G(y) \\ &:= \int F(z - y) dG(y). \end{aligned} \quad (1.3)$$

As the proofs of Theorem 1.4 and Theorem 1.5 require additional lemmas and definitions which are not in the scope of this thesis, we refer to [20] for the proofs. The right hand side of Eq. (1.3) is called the *convolution* of  $F$  and  $G$ , where  $*$  denotes the convolution operator. Theorem 1.5 provides us a way to obtain a sum of independent random variables. For an efficient computation of the convolution the following theorem can be applied ([9]).

**Theorem 1.6.** Let  $\mathcal{F}\{f\}$  and  $\mathcal{F}\{g\}$  be the Fourier transforms of  $f$  and  $g$ , respectively. Then

$$\mathcal{F}\{f * g\} = \mathcal{F}\{f\} \cdot \mathcal{F}\{g\}.$$

*Proof.* We have

$$f(t) = \mathcal{F}^{-1}[F(\nu)](t) = \int F(\nu) e^{2\pi i \nu t} d\nu$$

$$g(t) = \mathcal{F}^{-1}[G(\nu)](t) = \int G(\nu) e^{2\pi i \nu t} d\nu,$$

where  $\mathcal{F}(t)$  denotes the inverse Fourier transform. The convolution is then

$$\begin{aligned}
 f * g &= \int g(t')f(t - t')dt' \\
 &= \int g(t') \left[ \int F(\nu)e^{2\pi i\nu(t-t')}d\nu \right] dt' \\
 &= \int F(\nu) \left[ \int g(t')e^{-2\pi i\nu t'}dt' \right] e^{2\pi i\nu t}d\nu \\
 &= \int F(\nu)G(\nu)e^{2\pi i\nu t}d\nu \\
 &= \mathcal{F}^{-1}[F(\nu)G(\nu)](t).
 \end{aligned}$$

□

## 1.3 Dependence

### 1.3.1 Copulas

Above we introduced independence between random variables. Here we discuss dependence, following [21] and [3]. Definitions, theorems, and propositions are based thereon, unless specified otherwise. For an  $n$ -dimensional vector of random variables  $(X_1, \dots, X_n)$  the dependence between  $X_1, \dots, X_n$  is described by the joint distribution function

$$F(x_1, \dots, x_n) = P(X_1 \leq x_1, \dots, X_n \leq x_n). \quad (1.4)$$

The concept of a copula is to separate  $F$  in Eq. (1.4) into one part describing the dependence structure and other parts describing solely the marginal behaviour. The next proposition is necessary to derive the methodology of copulas.

**Proposition 1.1.** *Let  $X$  be a random variable with distribution function  $F$ . Let  $F^{-1}$  be the quantile function of  $F$ , i.e.*

$$F^{-1}(\alpha) = \inf\{x | F(x) \geq \alpha\},$$

with  $\alpha \in (0, 1)$ . Then

- (i) *For any standard-uniform distributed  $U \sim U(0, 1)$  we have  $F^{-1}(U) \sim F$ , providing a simple method to simulate random variables with distribution function  $F$ .*

- (ii) If  $F$  is continuous then the random variable  $F(X)$  is standard-uniformly distributed, i.e.  $F(X) \sim U(0, 1)$ .

In order to achieve the separation of  $F$  into the respective parts, we first transform each component of the random vector  $\mathbf{X} = (X_1, \dots, X_n)^T$  to have standard-uniform marginal distributions  $U(0, 1)$ . This can be accomplished by using the probability-integral transformation  $T : \mathcal{R}^n \rightarrow \mathcal{R}^n, (x_1, \dots, x_n)^T \rightarrow (F_1(x_1), \dots, F_n(x_n))^T$ , assuming that  $X_1, \dots, X_n$  have continuous marginal distributions  $F_1, \dots, F_n$ . Then, the joint distribution function  $C$  is referred to as the copula of  $(X_1, \dots, X_n)^T$ . This sets the basis for Sklar's theorem.

**Definition 1.12.** A *copula* is the distribution function of a random vector in  $\mathcal{R}^n$  with uniform-(0,1) marginals. Alternatively, a copula is any function  $C : [0, 1]^n \rightarrow [0, 1]$ , which has the following properties:

- (i)  $C(x_1, \dots, x_n)$  is increasing in each component  $x_i$ .
- (ii)  $C(1, \dots, 1, x_i, 1, \dots, 1) = x_i$  for all  $i \in \{1, \dots, n\}, x_i \in [0, 1]$ .
- (iii) For all  $(a_1, \dots, a_n), (b_1, \dots, b_n) \in [0, 1]^n$  with  $a_i \leq b_i$  it holds:

$$\sum_{i_1=1}^2 \dots \sum_{i_n=1}^2 (-1)^{i_1+\dots+i_n} C(x_{1i_1}, \dots, x_{ni_n}) \geq 0,$$

where  $x_{j1} = a_j$  and  $x_{j2} = b_j$  for all  $j \in \{1, \dots, n\}$ .

**Theorem 1.7** (Sklar's theorem). *Let  $F$  be an  $n$ -dimensional distribution function with margins  $F_1, F_2, \dots, F_n$ . Then there exists an  $n$ -dimensional Copula  $C$  such that for all  $(x_1, \dots, x_n)$  in  $\mathbb{R}^n$*

$$F(x_1, x_2, \dots, x_n) = C(F_1(x_1), \dots, F_n(x_n)). \quad (1.5)$$

*For any continuous multivariate distribution  $C$  is unique. Conversely, if  $F_1, \dots, F_n$  are not all continuous, then  $C$  is not unique.*

The proof of Theorem 1.7 can be found in [3]. Sklar's theorem provides an efficient way of modelling dependent random variables. Following from Eq. (1.5) the joint density function for a  $n$ -dimensional vector  $(x_1, \dots, x_n) \in \mathbb{R}^n$  can be obtained by

$$f(x_1, \dots, x_n) = c(F_1(x_1), \dots, F_n(x_n)) \prod_{i=1}^n f_i(x_i) \quad (1.6)$$

with copula density

$$c(F_1(x_1), \dots, F_n(x_n)) = \frac{\partial C(F_1(x_1), \dots, F_n(x_n))}{\partial F_1(x_1), \dots, \partial F_n(x_n)}.$$

Copulas can be classified into two major categories - elliptical and archimedian copulas. The first class is derived from its related elliptical distributions. In the following we describe the copulas, which will be used throughout this thesis. The Gaussian copula is an elliptical copula, which is derived from the joint normal distribution

$$\Phi_{\Sigma}(\mathbf{x}) = \frac{1}{(2\pi)^{(n/2)}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

for  $\mathbf{x} = (x_1, \dots, x_n)^T \in \mathbb{R}$  with mean vector  $\boldsymbol{\mu}$  and covariance matrix  $\Sigma$ . The  $n$ -dimensional Gaussian copula is given by

$$C^{\text{Gauss}}(u_1, \dots, u_n) = \Phi_{\Sigma}(\Phi^{-1}(u_1), \dots, \Phi^{-1}(u_n)), \quad (1.7)$$

where  $u_i := F_i(x_i)$ ,  $i \in (1, \dots, n)$  and  $\Phi^{-1}$  is the quantile function of the univariate standard normal distribution. Another elliptical copula is based on the Student-t distribution

$$t_{\nu, \Sigma}(\mathbf{x}) = \frac{\Gamma(\frac{\nu+n}{2})}{\Gamma(\frac{\nu}{2})\sqrt{(\pi\nu)^n|\Sigma|}} \left(1 + \frac{(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})}{\nu}\right)^{-\frac{\nu+n}{2}},$$

thus, referred to as Student-t copula, with *gamma function*  $\Gamma(x) = (x-1)!$ . It is defined analogous to the Gaussian copula in Eq. (1.7).

Archimedian copulas are not derived from parametric distribution. Instead, they are based on a *generator function*  $\varphi$ . Archimedian copulas are defined as

$$C^{\text{Archimedian}}(u_1, \dots, u_n) = \varphi^{(-1)}(\varphi(u_1) + \dots + \varphi(u_n)),$$

where  $\varphi^{(-1)}$  is the pseudo-inverse of  $\varphi$ . Archimedean copulas are popular in empirical applications as they produce wide ranges of dependence properties for different choices of the generator function. Examples of Archimedian copulas are the Clayton Copula with the generator function

$$\varphi^{\text{Clayton}}(u) = (1+u)^{(-1/\alpha)},$$

and the Gumbel Copula with the generator function

$$\varphi^{\text{Gumbel}}(u) = \exp(-u^{(-1/\alpha)}),$$

where  $\alpha$  is the Copula parameter, which describes the dependency between the random variables  $x_i$ .

### 1.3.2 Correlation

The terms correlation and dependence are often set as equal, however, correlation is only one measure of stochastic dependence. More precisely, [21] defines correlation as "the canonical measure in the world of multivariate normal distributions, and more generally for spherical and elliptical distributions".

**Definition 1.13.** The *linear correlation coefficient*, also known as *Pearson's correlation coefficient*, between two random variables  $X$  and  $Y$  is given by

$$\rho(X, Y) = \frac{\text{Cov}[X, Y]}{\sqrt{\sigma^2[X]\sigma^2[Y]}},$$

where  $\text{Cov}[X, Y]$  is the covariance between  $X$  and  $Y$ ,  $\text{Cov}[X, Y] = E[XY] - E[X]E[Y]$ , and  $\sigma^2[X], \sigma^2[Y]$  denote the variances of  $X$  and  $Y$ .

As its name suggests, the linear correlation coefficient is a measure of linear dependence. If  $X$  and  $Y$  are independent, then  $\rho(X, Y) = 0$ . If  $X$  and  $Y$  are perfectly linear dependent, i.e.  $Y = aX + b$  for  $a \in \mathbb{R} \setminus \{0\}$ ,  $b \in \mathbb{R}$ , then  $\rho(X, Y) = \pm 1$ . This follows from

$$\rho(X, Y)^2 = \frac{\sigma^2[Y] - \min_{a,b} E[(Y - (aX + b))^2]}{\sigma^2[Y]}. \quad (1.8)$$

If  $X$  and  $Y$  are imperfectly linear dependent, then  $-1 < \rho(X, Y) < 1$ . Linearity property is fulfilled by correlation, i.e.

$$\rho(\alpha X + \beta, \gamma Y + \delta) = \text{sgn}(\alpha \cdot \gamma) \rho(X, Y),$$

for  $\alpha, \gamma \in \mathbb{R} \setminus \{0\}$ ,  $\beta, \delta \in \mathbb{R}$ . Thus, correlation is invariant under strictly increasing linear transformations.

A correlation measure gives us the opportunity to describe the dependence structure of two random variables with a single number. The desired properties that a dependence measure  $\delta(\cdot, \cdot)$  should fulfil are the following:

P1. Symmetry:  $\delta(X, Y) = \delta(Y, X)$ .

P2. Normalisation:  $-1 \leq \delta(X, Y) \leq 1$ .

P3.  $X, Y$  comonotonic  $\Leftrightarrow \delta(X, Y) = 1$ ;  
 $X, Y$  countermonotonic  $\Leftrightarrow \delta(X, Y) = -1$ .

P3. For  $T : \mathbb{R} \rightarrow \mathbb{R}$  strictly monotonic on the range of  $X$ :

$$\delta(T(X), Y) = \begin{cases} \delta(X, Y) & T \text{ increasing} \\ -\delta(X, Y) & T \text{ decreasing.} \end{cases} \quad (1.9)$$

In the case of linear correlation as a dependence measure only P1 and P2 are fulfilled. Another measurement of correlation is rank correlation. There are two rank correlation coefficients.

**Definition 1.14.** Let  $X, Y$  be random variables with distribution functions  $F_1$  and  $F_2$ , respectively, and joint distribution function  $F$ . *Spearman's rank correlation*, also referred to as *Spearman's rho*, is given by

$$\rho_s(X, Y) = \rho(F_1(X), F_2(Y)), \quad (1.10)$$

where  $\rho$  is Pearson's linear correlation coefficient.

**Definition 1.15.** Let  $(X_1, Y_1), (X_2, Y_2)$  be two independent pairs of random variables from  $F$ , then *Kendall's rank correlation*, also referred to as *Kendall's tau*, is given by:

$$\tau(X, Y) = P[(X_1 - X_2)(Y_1 - Y_2) > 0] - P[(X_1 - X_2)(Y_1 - Y_2) < 0]. \quad (1.11)$$

The advantage of rank correlation is that it measures the degree of monotonic dependence between  $X$  and  $Y$  instead of only measuring linear dependence.

**Theorem 1.8.** Let  $X$  and  $Y$  be random variables with distributions  $F_1$  and  $F_2$ , respectively, joint distribution  $F$  and copula  $C$ . Then the following holds:

- (i)  $\rho_s(X, Y) = \rho_s(Y, X), \tau(X, Y) = \tau(Y, X)$ .
- (ii) If  $X$  and  $Y$  are independent then  $\rho_s(X, Y) = \tau(X, Y) = 0$ .
- (iii)  $-1 \leq \rho_s(X, Y), \tau(X, Y) \leq 1$ .
- (iv)  $\rho_s(X, Y) = 12 \int_0^1 \int_0^1 \{C(x, y) - xy\} dx dy$ .
- (v)  $\tau(X, Y) = 4 \int_0^1 \int_0^1 C(u, v) dC(u, v) - 1$ .

(vi) For  $T : \mathbb{R} \rightarrow \mathbb{R}$  strictly monotonic on the range of  $X$ , both  $\rho_s$  and  $\tau$  satisfy Eq. (1.9).

(vii)  $\rho_s(X, Y) = \tau(X, Y) = 1 \Leftrightarrow C = C_u \Leftrightarrow Y = T(X)$  a.s. with  $T$  increasing.

(viii)  $\rho_s(X, Y) = \tau(X, Y) = -1 \Leftrightarrow C = C_l \Leftrightarrow Y = T(X)$  a.s. with  $T$  decreasing.

*Proof.* The verification of (i), (ii) and (iii) are straightforward. (iv) Using the identity due to [30]

$$\text{Cov}[X, Y] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \{F(x, y) - F_1(x)F_2(y)\} dx dy \quad (1.12)$$

which can be found in [16].

(v) Calculate

$$\begin{aligned} \tau(X, Y) &= 2P[(X_1 - X_2)(Y_1 - Y_2) > 0] - 1 \\ &= 2 \cdot 2 \iiint_{\mathbb{R}^4} \mathbf{1}_{\{x_1 > x_2\}} \mathbf{1}_{\{y_1 > y_2\}} dF(x_2, y_2) dF(x_1, y_1) - 1 \\ &= 4 \iint_{\mathbb{R}^2} F(x_1, y_1) dF(x_1, y_1) - 1 \\ &= 4 \iint C(u, v) dC(u, v) - 1. \end{aligned}$$

(vi) Is a consequence of the fact that  $\tau$  and  $\rho_s$  can be expressed in terms of the copula which is invariant under strictly increasing transformations of the marginals.

(vii) From (iv) it follows that  $\rho_s(X, Y) = 1$  is maximized if only if  $C = C_u$  if and only if  $Y = T(X)$  a.s. Suppose  $Y = T(X)$  a.s. with  $T$  increasing, then due to the continuity of  $F_2$  it follows that

$$P[Y_1 = Y_2] = P[T(X_1) = T(X_2)] = 0,$$

which means

$$\tau(X, Y) = P[Y_1 = Y_2] = P[(X_1 - X_2)(Y_1 - Y_2) > 0] = 1.$$

Conversely,  $\tau(X, Y) = 1$  implies

$$P \times P[(\omega_1, \omega_2) \in \Omega \times \Omega | (X(\omega_1) - X(\omega_2))(Y(\omega_1) - Y(\omega_2)) > 0] = 1.$$

Let  $A, B$  be sets with  $A = \{\omega \in \Omega | X(\omega) \leq x\}$  and  $B = \{\omega \in \Omega | Y(\omega) \leq y\}$ . Assuming  $P[A] \leq P[B]$ , we need to show  $P[A \cap B] = P[A]$ . If  $P[A \setminus B] > 0$

then also  $P[B \setminus A] > 0$  and

$$(X(\omega_1) - X(\omega_2))(Y(\omega_1) - Y(\omega_2)) < 0$$

on the set  $(A \setminus B) \times (B \setminus A)$ , which has measure  $P[A \setminus B] \cdot P[B \setminus A] > 0$ , which is a contradiction. Hence  $P[A \setminus B] = 0$ , thus,  $P[A \cap B] = P[A]$ .

(viii) Can be proven analogously to (vii).  $\square$

With this result we have shown that rank correlation has properties P1, P2, P3 and P4.

### 1.3.3 Dependent Discrete Convolution

Here, we present the methodology of the Dependent Discrete Convolution (DDC) ([70, 63]). We denote travel time for segment 1 as a random variable  $\alpha$ , and travel time for segment 2 as  $\beta$ . The PDF for path travel time distribution denoted as  $f_{\alpha+\beta}$  can be calculated from the joint Probability Density Function (PDF)  $f_{\alpha,\beta}$  with Cumulative Distribution Function (CDF)  $F_\alpha$  and  $F_\beta$ , and PDFs  $f_\alpha$  and  $f_\beta$ , respectively:

$$f_{\alpha+\beta}(a) = \int_{\underline{\alpha}}^{\bar{\alpha}} f_{\alpha,\beta}(x, a-x) dx, \quad a \in [\underline{\alpha} + \underline{\beta}, \bar{\alpha} + \bar{\beta}],$$

where  $[\underline{\alpha}, \bar{\alpha}]$  is the domain of  $\alpha$  and  $[\underline{\beta}, \bar{\beta}]$  is the domain of  $\beta$ . If  $\alpha$  and  $\beta$  are independent,  $f_{\alpha,\beta}(x, y)$  can be obtained by the product of its marginal PDFs:

$$f_{\alpha,\beta}(x, y) = f_\alpha(x) f_\beta(y),$$

and 1.3.3 becomes

$$f_{\alpha+\beta}(a) = \int_{\underline{\alpha}}^{\bar{\alpha}} f_\alpha(x) f_\beta(a-x) dx, \quad a \in [\underline{\alpha} + \underline{\beta}, \bar{\alpha} + \bar{\beta}],$$

which is the convolution equation. For computerized calculation, the discrete form of convolution is used:

$$P_{\alpha+\beta}(i) = \sum_{j=i_{\underline{\alpha}}}^{i_{\bar{\alpha}}} P_\alpha(j) P_\beta(i-j), \quad i \in [i_{\underline{\alpha}} + i_{\underline{\beta}}, i_{\bar{\alpha}} + i_{\bar{\beta}}],$$

where  $P_\alpha(i)$ ,  $P_\beta(i)$  and  $P_{\alpha+\beta}(i)$  are the discrete approximations of  $f_\alpha(\cdot)$ ,  $f_\beta(\cdot)$  and  $f_{\alpha+\beta}(a)$ . For that purpose, an  $N$ -dimensional space is discretized with



a step of  $\Delta p$ , as shown in Figure 1.1. For each dimension  $\Delta p$  may differ, however, for simplicity reasons the same  $\Delta p$  is chosen for every dimension.

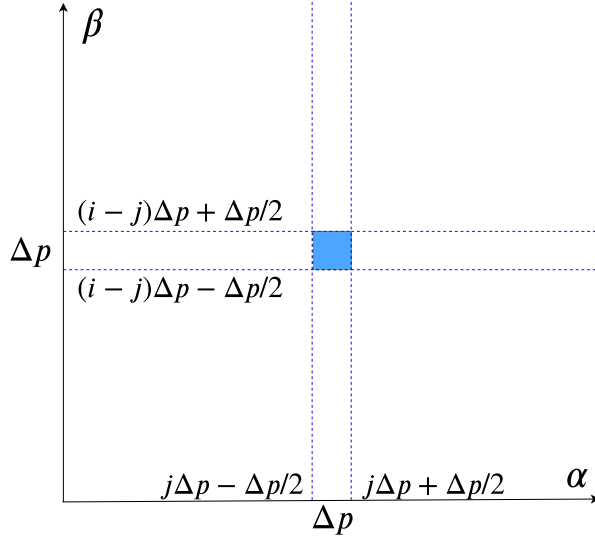


FIGURE 1.1: Discretization of the 2-dimensional plane for DDC.

Then the discrete approximation of  $f_\alpha(\cdot)$  is defined as follows:

$$P_\alpha(i) = \int_{i\Delta p - \Delta p/2}^{i\Delta p + \Delta p/2} f_\alpha(x) dx.$$

The discrete approximation  $P_\beta(i)$  of  $f_\beta(\cdot)$  is defined analogously.

Now we consider the case, where  $\alpha$  and  $\beta$  are dependent. Then (1.3.3) does not hold. However, based on Sklar's theorem we can use (1.6) and obtain

$$f_{\alpha,\beta}(x, y) = c(F_\alpha(x), F_\beta(y)) f_\alpha(x) f_\beta(y).$$

Using (1.3.3) and (1.3.3), the dependent convolution becomes

$$f_{\alpha+\beta}(a) = \int_{\underline{\alpha}}^{\bar{\alpha}} c(F_\alpha(x), F_\beta(a-x)) f_\alpha(x) f_\beta(a-x) dx.$$

Then the DDC can be formulated as

$$\begin{aligned} P_{\alpha+\beta}(i) &= \sum_{j=i_{\underline{\alpha}}}^{i_{\bar{\alpha}}} \int_{j\Delta p - \Delta p/2}^{j\Delta p + \Delta p/2} \int_{(i-j)\Delta p - \Delta p/2}^{(i-j)\Delta p + \Delta p/2} f_{\alpha,\beta} dx dy \\ &= \sum_{j=i_{\underline{\alpha}}}^{i_{\bar{\alpha}}} \int_{j\Delta p - \Delta p/2}^{j\Delta p + \Delta p/2} \int_{(i-j)\Delta p - \Delta p/2}^{(i-j)\Delta p + \Delta p/2} c(F_\alpha(x), F_\beta(y)) f_\alpha(x) f_\beta(y) dx dy. \end{aligned}$$

If  $\Delta p$  is small enough,  $c(F_\alpha(x), F_\beta(y))$  can be approximated as a constant over the 2-dimensional interval of  $[j\Delta p - \Delta p/2, j\Delta p + \Delta p/2] \times [(i-j)\Delta p - \Delta p/2, (i-j)\Delta p + \Delta p/2]$  and is denoted as  $c_{j\Delta p, (i-j)\Delta p}$ . It can be calculated by

$$c_{j\Delta p, (i-j)\Delta p} = c \left( \sum_{m=0}^j P_\alpha(m), \sum_{n=0}^{i-j} P_\beta(n) \right).$$

Then (1.3.3) becomes

$$\begin{aligned} P_{\alpha+\beta}(i) &= \sum_{j=i_\alpha}^{i_\alpha} c_{j\Delta p, (i-j)\Delta p} \int_{j\Delta p - \Delta p/2}^{j\Delta p + \Delta p/2} f_\alpha(x) dx \int_{(i-j)\Delta p - \Delta p/2}^{(i-j)\Delta p + \Delta p/2} f_\beta(y) dy, \\ &= \sum_{j=i_\alpha}^{i_\alpha} c_{j\Delta p, (i-j)\Delta p} P_\alpha(j) P_\beta(i-j). \end{aligned}$$

Compared to the traditional convolution the DDC has similar form as (1.3.3), except that there is a an extra multiplier  $c_{j\Delta p, (i-j)\Delta p}$  determined by the dependency between  $\alpha$  and  $\beta$ . When  $\alpha$  and  $\beta$  are independent,  $c_{j\Delta p, (i-j)\Delta p} := 1$  and (1.3.3) degenerates into the traditional convolution in (1.3.3). In case of dependent segments, this factor incorporates the copula based correlation between the segments.

## Chapter 2

# Shortest Path Problem and Data Sampling

## 2.1 Shortest Path Problem

Route guidance systems commonly aim at finding the shortest route for a specified origin – destination pair. Finding the shortest route can be formulated as a Shortest Path Problem (SPP), which is a well-studied combinatorial optimization problem.

In this chapter, we first present the SPP and the state-of-the-art solution approach. Next, we discuss how we can obtain alternate routes for one  $o - d$  pair instead of only the shortest route.

### 2.1.1 Problem Formulation

The road network is represented as a directed graph  $G(N, L, C)$ , where  $N$  is the set of nodes ( $|N| = n$ ) representing geolocations,  $L$  ( $|L| = m$ ) is the set of links representing road links connecting these locations, and  $C$  ( $|C| = m$ ) contains the link costs representing travel times on each road link. An example of a directed graph is illustrated in Figure 2.1. The goal of finding the shortest path from origin  $o \in N$  to destination  $d \in N$  can be achieved by solving the SPP.

State-of-the-art route guidance systems underlie a deterministic SPP where the estimated link travel times are discrete. Travel times can be assessed using real-time traffic information in combination with historical data. We will give a detailed explanation on how to obtain historical travel time data in Chapter 5. There we will also discuss non-deterministic SSPs.

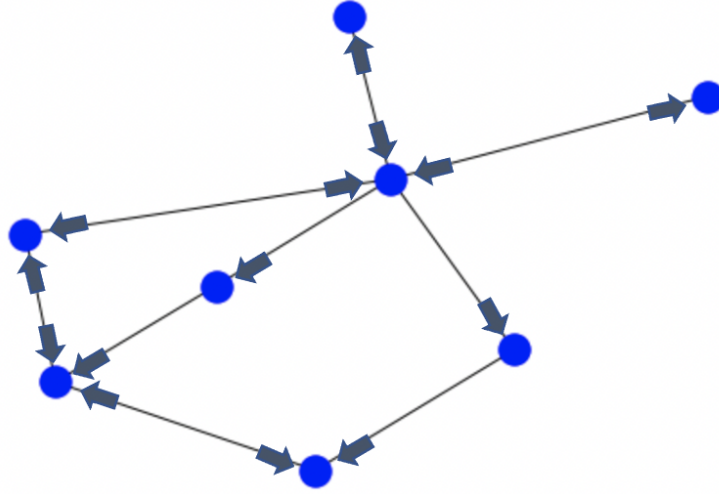


FIGURE 2.1: A directed Graph with blue nodes and black links.  
The arrows represent the possible directions for each link.

### 2.1.2 Dijkstra's Algorithm

Dijkstra's Algorithm, first proposed in [18], provides an efficient method to solve the SSP. Its description, based on [18] and [23], can be found in Algorithm 1. Worst case time complexity of Dijkstra's Algorithm is  $O(N^2)$ , however using a Fibonacci heap it becomes  $O(L + N \log N)$ . Following [18] we will

---

**Algorithm 1:** Dijkstra ( $G, s$ )

---

```

 $\forall t \in N, d[t] \leftarrow \infty$  // set initial distance estimates ;
 $d[s] \leftarrow 0$  ;
 $F \leftarrow \{v | \forall v \in V\}$  //  $F$  is the set of nodes that are yet to achieve final
    distance estimates ;
 $D \leftarrow \emptyset$  //  $D$  will be the set of nodes that have achieved final distance
    estimates ;
while  $F \neq \emptyset$  do
     $x \leftarrow$  element in  $F$  with minimum distance estimate ;
    for  $(x, y) \in E$  do
         $d[y] \leftarrow \min\{d[y], d[x] + c(x, y)\}$  // relax the estimate of  $y$  to
        maintain paths: if  $d[y]$  changes, then  $\pi(y) \leftarrow x$ ;
    end
     $F \leftarrow F \setminus \{x\}$ ;
     $D \leftarrow D \cup \{x\}$ 
end

```

---

prove that Dijkstra's algorithm calculates the distances from  $s$  to all  $t \in V$  correctly.

**Proposition 2.1.** *For every  $u$ , at any point of time  $d[u] \geq d(s, u)$ .*

*Proof.* Follows by induction by showing that any point in time, if  $d[u] < \infty$ , then  $d[u]$  is the weight of some path from  $s$  to  $u$ . Hence at any point  $d[u]$  is at least the weight of the shortest path, and thus  $d[u] \geq d(s, u)$ . As  $d[s] = 0 = d(s, s)$  and all other distance estimates are  $+\infty$  the claim holds initially. When  $d[u]$  is changed to  $d[x] + c(x, u)$  then there exists a path from  $s$  to  $x$  with cost  $d[x]$  and an edge  $(x, u)$  with cost  $c(x, u)$ . Subsequently there is a path from  $s$  to  $u$  with cost  $d[u] = d[x] + c(x, u)$ . This implies that  $d[u]$  is at least the cost of the shortest path  $d(s, u)$ , which completes the induction argument.  $\square$

**Proposition 2.2.** *When node  $x$  is placed in  $D$ ,  $d[x] = d(s, x)$ .*

*Proof.* Follows by induction on the order of placements of nodes into  $D$ . In the basic case  $s$  is placed into  $D$  where  $d[s] = d(s, s) = 0$ , thus, the claim holds initially. For the inductive step we assume that for all nodes  $y$  currently in  $D$ ,  $d[y] = d(s, y)$ . Let  $x$  be the node with minimum distance estimate in  $F$  at the moment. In order to complete the induction we will show that  $d[x] = d(s, x)$ . Let  $p$  be the shortest path from  $s$  to  $x$ . Suppose  $z$  is the node on  $p$  which is closest to  $x$  with  $d[z] = d(s, z)$ . Since  $d[s] = d(s, s)$  we know  $z$  exists. For every node  $y$  on  $p$  between  $z$  to  $x$  by the choice of  $z$  it holds that  $d[y] > d(s, y)$ . We consider the following options for  $z$ .

1. If  $z = x$ , then  $d[x] = d(s, x)$ .
2. Suppose  $z \neq x$ . Then there exists a node  $z'$  after  $z$  on  $p$ . It holds that  $d[z] = d(s, z) \leq d(s, x) \leq d[x]$  as subpaths of shortest paths are also shortest paths, so that the prefix of  $p$  from  $s$  to  $z$  has cost  $d(s, z)$ . Furthermore, the costs on links are non-negative, so that the portion of  $p$  from  $z$  to  $x$  has a non-negative cost, hence,  $d(s, z) \leq d(s, x)$ . Now, suppose  $d[z] < d[x]$ . By the choice of  $x \in F$  it holds that  $d[x]$  is the minimum distance estimate that was in  $F$ . Hence, since  $d[z] < d[x]$ ,  $z \notin F$  has to be in  $D$ . Subsequently,  $d[z'] \leq d(s, z) + c(z, z')$ , as  $z$  lies on the shortest path from  $s'$  to  $z$ , and the distance estimate of  $z'$  must be correct. But this contradicts  $z$  being the closest node on  $p$  to  $x$  fulfilling  $d[z] = d(s, z)$ . Hence, our initial assumption that  $d[z] < d[x]$  has to be false which leads to  $d[z] = d(s, x)$ .

$\square$

Note that the correctness of Dijkstra's algorithm follows from Proposition 2.2 since  $d[x]$  remains unchanged after  $x$  is added to  $D$ . The only possibility

to change it would be for some node  $y \in F$ ,  $d[y] + c(y, x) < d[x]$  which is not possible since  $d[x] \leq d[y]$  and  $c(y, x) \geq 0$ . For all  $y \in F$  it holds that  $d[x] \leq d[y]$  at all points after  $x$  is inserted into  $D$ .

### 2.1.3 Yen's $k$ -shortest Path Algorithm

The  $k$ -shortest path problem is a generalization of the SPP which aims at computing  $k$  paths from origin to destination in non-decreasing order of their costs. We will use Yen's algorithm, which was introduced in [67], in order to find the  $k$ -shortest paths.

We will describe Yen's Algorithm based on [51]. First, the shortest path from  $s$  to  $d$ , denoted as  $P_0 = (s, \dots, t)$ , is computed using Dijkstra's Algorithm. A priority queue  $Q$  of candidates for the upcoming shortest paths is maintained. At the start of the algorithm,  $C$  consists of only  $P_0$  with the priority weight( $P_0$ ). Then, the shortest paths  $P_i, i \in \{1, \dots, k-1\}$ , are computed and  $C$  is updated with candidate paths for the upcoming iteration. We obtain path  $P_i$  by extracting the minimum weight path from  $C$ . Let  $P_i = (v_1^i = s, v_2^i, \dots, v_d^i, \dots, v_l^i = t)$  and let  $P_j$  be the path with the maximum common prefix  $(v_1^j = v_1^i, v_2^j = v_2^i, \dots, v_d^j = v_d^i)$  among the total set of paths  $\{P_1, \dots, P_{i-1}\}$ . In addition, let  $dev(P_i) = v_d^i$  be the node at which  $P_i$  diverged from  $P_j$ . The shortest path deviating from  $P_i$  at nodes  $\{v_d^i, \dots, v_{l-1}^i\}$  are then the new candidate paths which will be inserted into  $C$ . In order to calculate the deviation path from a node  $v_f^i$ , the shortest path  $P^f$  is first computed from  $v_f^i$  to  $t$  in the graph  $G'$  obtained by removing nodes  $\{v_1^i, \dots, v_{f-1}^i\}$ , incident links and all links in  $E^f$  from  $G$ , where  $E^f = \{(v_f^h, v_{f+1}^h) : 1 \leq h \leq i \text{ and } v_1^h = v_1^i, \dots, v_f^h = v_f^i\}$ . The shortest deviation path from  $P_i$  at node  $v_f^i$  is the path  $\{v_1^i, v_2^i, \dots, v_{f-1}^i\}$  appended by  $P^f$ , which is inserted into  $C$  with its weight as priority. Algorithm 2 gives a pseudo-code for the algorithm. Dijkstra's Algorithm is performed  $kv$  times within Yen's Algorithm, where  $v$  is the length of spurs paths. As the expected value of  $v$  is  $O(\log N)$ , time complexity becomes  $O(KN(M + N \log N))$ . There is no additional theoretical basis concerning Yen's Algorithm as it relies on the underlying theory of Dijkstra's Algorithm.

**Algorithm 2:** Yen  $(G, s, d), K$ 


---

```

 $K_{\text{paths}} \leftarrow \emptyset ;$ 
 $P_0 \leftarrow \text{Dijkstra}(G, s, d) ;$ 
 $\alpha_0 \leftarrow 0 ;$ 
 $j_0 \leftarrow 0 ;$ 
 $k \leftarrow 1 ;$ 
while  $k < K$  and  $Q \neq \emptyset$  do
     $P_{\text{root}} = (v_0^{k-1}, \dots, v_{\alpha_{k-1}-1}^{k-1}) ;$ 
     $G'(V', E') = G[V \setminus P_{\text{root}}]$  for  $i \in \{0, \dots, k-2\}$  do
        if  $j_i == j_{k-1}$  or  $i == j_{k-1}$  then
             $E' = E' \setminus \{(v_{\alpha_i}^i, v_{\alpha_i+1}^i)\}$ 
        end
    end
    for  $n = \alpha_{k-1}, \dots, q_{k-1} - 1$  do
         $E' = E' \setminus \{(v_n^{k-1}, v_{n+1}^{k-1})\} ;$ 
         $P = \text{Dijkstra}(G', n) ;$ 
         $P = P_{\text{root}} \cap P, j = k-1, \alpha = n ;$ 
         $Q = Q \cup \{(P, j, \alpha)\} ;$ 
         $G'(V', E') = G[V \setminus \{v_n^{k-1}\}] ;$ 
    end
     $(P_k, j_k, \alpha_k) = \text{shortest path in } Q ;$ 
     $Q = Q \setminus \{P_k, j_k, \alpha_k\} ;$ 
     $K_{\text{paths}} = K_{\text{paths}} \cup \{P_k\} ;$ 
     $k = k + 1 ;$ 
end

```

---

## 2.2 Data

In order to solve the SPP link costs are necessary representing travel times for each link. These can be estimated in route guidance systems using real time traffic information. Another possibility is to collect historical travel time data. This approach has several advantages and is crucial for developing a stochastic router, which we will be described later in this thesis. There are multiple possibilities to obtain historical travel time data. We will give an overview in the next chapter, when we will discuss related work.

Here we rather focus on presenting the so called Floating Car Data (FCD) which will be used throughout this thesis. First, we give an overview on how the the raw data is sampled and collected. Then, we elucidate how the raw data is processed in order to obtain link travel time data.

### 2.2.1 Sampling and Collection

The FCD provided by BMW Group is collected from probe vehicles. The setup includes a fleet of probe vehicles, which has a module that reports Global Navigation Satellite System (GNSS) data and a central server, which collects all data in a database. A GNSS position is determined via the principle of trilateration using satellite signals as illustrated in Figure 2.2. Each vehicle samples the current GNSS positions in intervals ranging from 10s to 30s, which are then stored in the local memory of the vehicle together with the according timestamp. After a few positions have been sampled, a filter mechanism decides whether sampled GNSS positions will be transmitted to the central server. This filter is continuously comparing the velocity of the vehicle with the velocity obtained by a traffic provider. If the velocity at one of the sampled data points differs more than 10 % to 30 % (depending on the software version on the module) from the provided velocity, the recently sampled positions and according timestamps are transmitted to the central server. Each transmitted position is linked to an alias, which is randomly generated by the vehicle and changes over time due to protection of the driver's privacy. At the central server, single transmitted positions of the same alias can be connected in order to reconstruct trajectories of the vehicles. However, since vehicles are not transmitting continuously, and hide their vehicle ID, it is not possible to reconstruct complete trips or infer driver's identity.



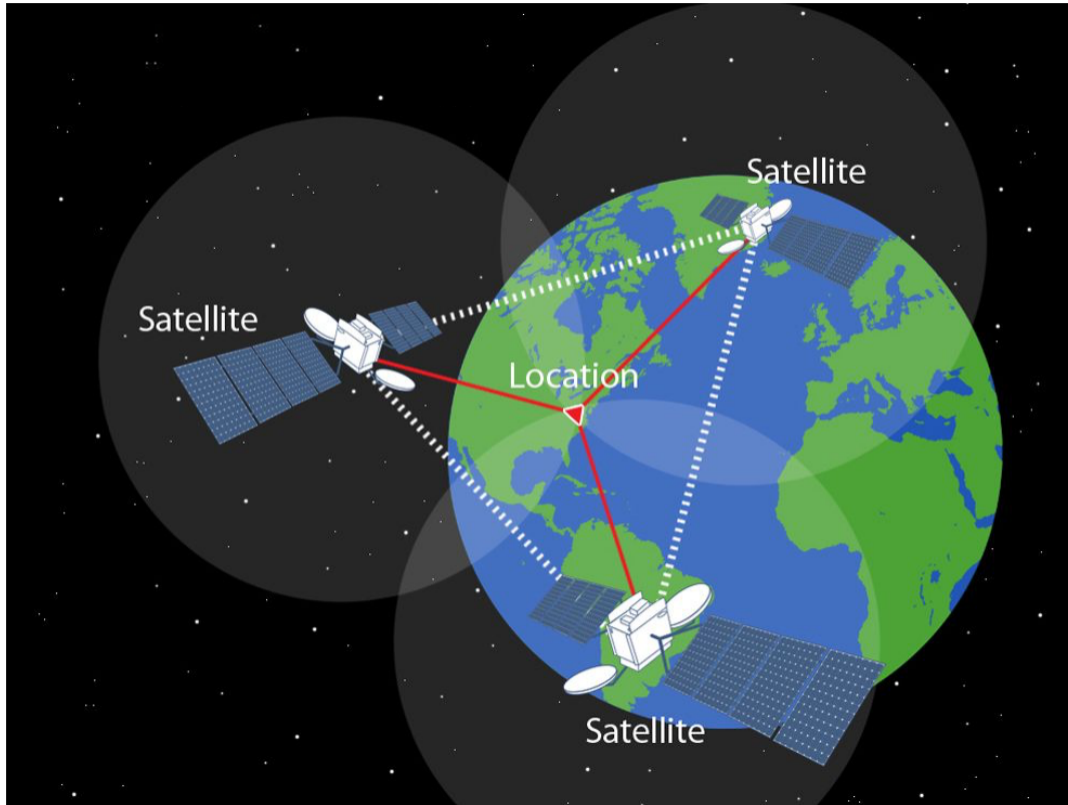


FIGURE 2.2: Determination of a GNSS position on earth using the principle of trilateration <sup>1</sup>.

### 2.2.2 Preprocessing

Raw data collected in the central server consist of GNSS positions and their according timestamps. This data then needs to be matched to links of the road network in a process called *map-matching*. This algorithm returns a sorted list of links depending on the position and time data which are supposed to match the roads that the respective vehicle traversed. Furthermore, functions are computed representing time-dependent position and velocity of the specific vehicle on the reconstructed link denoted as  $x_c(t)$  and  $v_c(t)$ , respectively. The map-matching algorithm is described in detail in [52]. A set of hypotheses of possible trajectories is constructed, which are all compared. Then, the most probable one is selected. The procedure for obtaining historical link travel times from probe vehicles is summarized in Figure 2.3.

### 2.2.3 Data Set

<sup>1</sup>Source: <https://www.nationalgeographic.org/photo/triangulation-sized>

<sup>2</sup>Source: <https://www.bmwgroup.com/en/brands-and-services/bmw.html>

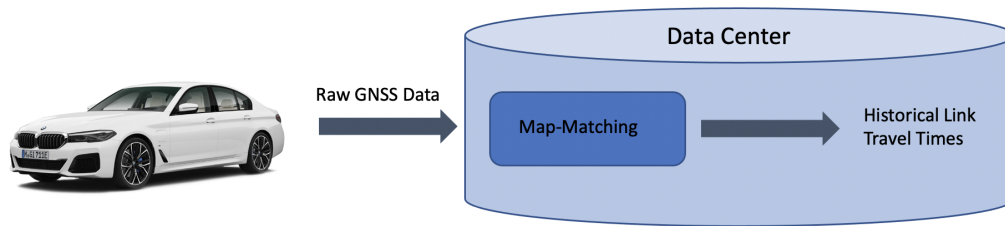


FIGURE 2.3: Obtaining historical link travel times from probe vehicles <sup>2</sup>.

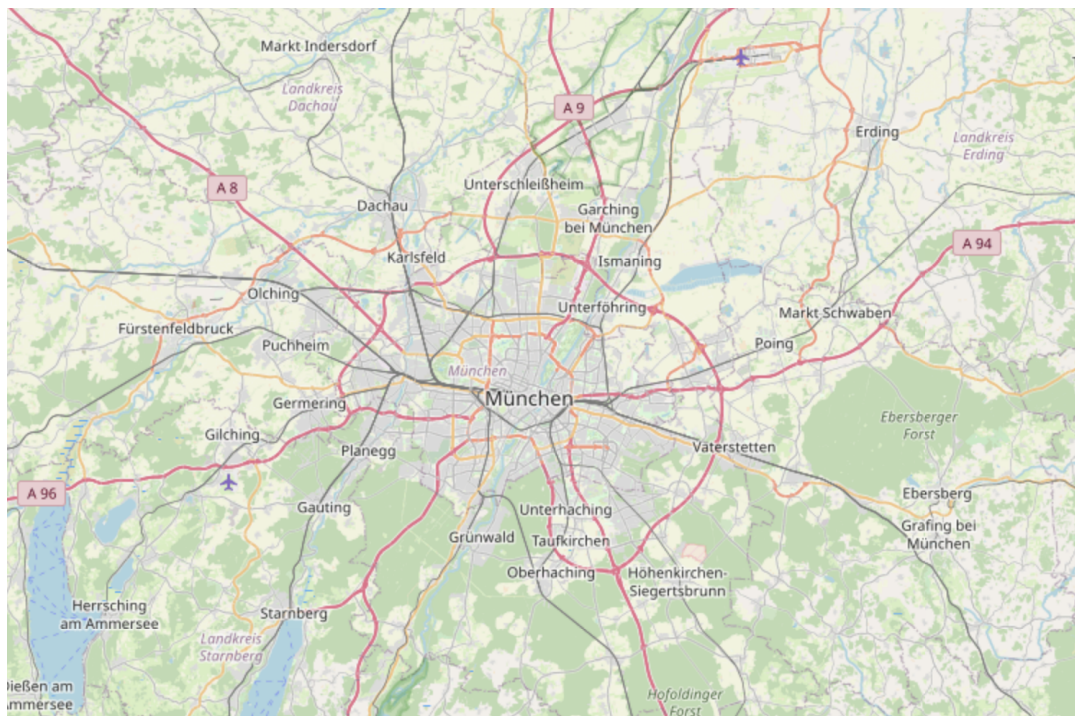


FIGURE 2.4: Map of Munich and its surroundings.

As maps as well as the software installed in the vehicles are constantly updated there are different data sets corresponding to the specific versions. In this thesis, we used a data set containing travel times for the area of Munich and its surroundings, which is shown in Figure 2.4. The data was collected during a period from March 2014 to March 2015.

## Chapter 3

# Modelling travel time reliability

### 3.1 Related Work

The US Federal Highway Administration [4] defined travel time reliability as the consistency or dependability in travel times, as measured from day to day and/or across different times of the day. So how can we measure travel time reliability? For that purpose a travel time distribution is required, which can be obtained from empirical data. The progress of information technology enables a variety of possibilities for travel time data collection. One example is to use loop detectors. ([54, 37, 38]). Another possibility to obtain data is Automated Vehicle Identification (AVI) ([13]). In [39] floating car data is used for travel time estimation. Probe vehicles can also be used as a source for travel time data ([49, 22, 35]). Travel time data is usually available only on link-level rather for longer paths consisting of multiple links. Historical travel times can then be used as an evaluation benchmark for statistical models estimating travel time distribution.

First we consider the state-of-the-art models for assessing link-level travel time distribution. In [68] a hierarchical Bayesian mixture model for assessing segment travel time distribution is developed based on bus probe data. Predominately bimodal distributions were revealed by the authors, with one mode corresponding to the uncongested state and the other corresponding to the congested state. In [19] an adaptive information fusion model was introduced which predicted short-term link travel time distribution by iteratively combining historical travel time data with real time information available at discrete time points. Furthermore, a partial differential equation describing link travel time distribution was proposed by [33]. The equation was solved in terms of Laplace transforms, which were inverted by a numerical inversion algorithm. Travel time distribution in large networks was assessed by [6] using the universal generating function. In [71] a probabilistic delay distribution

model with stochastic arrivals and departures was presented for investigating urban link travel time distribution.

In order to enable providing route guidance it is necessary to determine path-level travel time distribution. A common approach is deducing path travel time distribution from its individual link travel time distributions. Most of the state-of-the-art methods for estimating link or path travel time distribution in a road network assume independence of individual link travel times ([31, 49]). But is this assumption realistic? Intuitively, one would assume that if one link is congested, then the successive link will very likely also be congested. This would induce a dependence of travel times. In [28] dependence between link travel times was found based on vehicle tracking data from Paramics simulation. It was shown that both temporal and spatial dependence between individual links needs to be taken into account for developing route guidance systems. In [46] the variance of freeway path travel time was assessed by modelling dependence between variances of link travel times. Here, the limitations are that the variances cannot fully describe the travel time properties. In [27] linear dependence between neighbouring link travel times was assumed for assessing the variance of one urban route travel time. However, link travel times may have complex dependence structure. In [47] Markov chains were used to predict path travel time. It was assumed that the traffic states of successive links form a Markov chain, but Markovian property of links is rather unrealistic. In [64] a regression approach was proposed for modelling travel time distributions from ambulances at trip level. It was argued, that the dependence between links is intrinsically included as the data was already available on path level. However, ambulances traversing with sirenes and flashing lights may not appropriately reflect the features and dependences of urban road travel times.

Recently, copulas, which are commonly used in econometrics ([60]), have been proposed for modelling dependence in path travel time distribution estimation ([13, 14, 50]). In [13] travel time distribution for a path consisting of two arterials in Shanghai, China and Los Angeles, California, based on AVI and Next Generation Simulation data. The Copula Model was compared to the empirical distribution and to the convolution method, which assumes independence between links. The results showed link travel times were dependent at both study sites, and that the Copula Model performed better than the convolution method. In [14] a Copula Model was developed and evaluated

by using VISSIM simulation with calibration to generate travel time data on one arterial in Hangzhou, China. The authors implemented two signal control strategies, denoted as "favorable" and "unfavorable". While they were able to determine different travel time states in the favorable case, no such distinction could be observed for the unfavorable case. Dependencies between successive links were analyzed in both cases. Also here, the Copula Model was then compared to the empirical distribution and to the convolution method. In the unfavorable case, no strong dependence could be measured, and the Copula Model showed no clear superiority over the convolution. But in the favorable case, a stronger dependence could be measured, and the Copula Model was closer to the empirical distribution than the convolution method.

As copulas are already a well-proven tool to model a variety of different dependence structures in econometrics and first promising results already occurred in the field of transportation systems, we will also use copulas in this thesis to model the dependence between link travel times for estimating path travel time distribution. However, there is still a long way to go from the state-of-the-art approaches mentioned above to an implementation based on copulas for real world application in route guidance systems. First, the benchmark for the validation of the the Copula Model in [13] and [14] does not represent real day-to-day travel time observations. Furthermore, travel time distribution was estimated for a path consisting of only two and three links, respectively. Thus, it is still an open question how the Copula Model performs with real day-to-day data and for an increased number of links.

In the following we tackle the open questions on the Copula Model by evaluating it using real day to day data and for an increased number of links. In addition, different copulas are applied, which where not yet used for estimating path travel time distribution. Eventually, we will expand the use of copulas for estimating travel time distribution beyond the Copula Model in order to improve efficiency for real world application.

## 3.2 Copula Model

First we give an overview of the methodology behind the Copula Model. Then, we conduct two case studies for different study sites for evaluation of the Copula Model. These two study sites show different road features, as the first one is a freeway arterial and the second one is an urban arterial. Different

road features may cause different dependence structures. We will go into detail of the study site description below.

### 3.2.1 Methodology

The Copula Model is comprised of two stages. At first, a continuous distribution for link travel times needs to be estimated from data. Then, the Copula Model can be fitted in order to obtain path travel time distributions in the second stage. We elucidate the procedure in the following. For computer implementation the R Framework ([57]) and the packages developed by [29], [65] and [34] were used.

The continuous distributions are estimated using the finite Gaussian Mixture Model (GMM) ([48]). For each segment travel time  $X_i$  we have a  $d$ -dimensional vector measurement  $\mathbf{x}_i$  obtained from the historical travel time dataset  $D = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ . We assume that the underlying density  $p_{\mathbf{x}}$  is defined as a finite mixture with  $K$  components:

$$p(\mathbf{x}|\Theta) = \sum_{k=1}^K \alpha_k p_k(\mathbf{x}|z_k, \theta_k)$$

where  $p_k(\mathbf{x}|z_k, \theta_k)$  are mixture components with parameters  $\theta_k, 1 \leq k \leq K$ ,  $z = (z_1, \dots, z_K)$  is a vector of  $K$  binary indicator variables that are mutually exclusive and exhaustive and represents the identity of the mixture component that generated  $\mathbf{x}$ , and  $\alpha_k = p(z_k)$  are the mixture weights, representing that the probability that a randomly selected  $\mathbf{x}$  was generated by component  $k$ , where  $\sum_{k=1}^K \alpha_k = 1$ . The complete set of parameters for a mixture model with  $K$  components is  $\Theta = \{\alpha_1, \dots, \alpha_K, \theta_1, \dots, \theta_K\}$ . The weight of data point  $\mathbf{x}$  in cluster  $k$ , given parameters  $\Theta$  is

$$\omega_{ik} = p(z_{ik} = 1|\mathbf{x}_i, \Theta) = \frac{p_k(\mathbf{x}_i|z_k, \theta_k) \cdot \alpha_k}{\sum_{m=1}^K p_m(\mathbf{x}_i|z_m, \theta_m) \cdot \alpha_m}, \quad (3.1)$$

where  $1 \leq k \leq K$  and  $1 \leq i \leq N$ . The weights reflect the uncertainty, given  $\mathbf{x}_i$  and  $\Theta$ , about which of the  $K$  components generated  $\mathbf{x}_i$ . For  $\mathbf{x}_i \in \mathbb{R}^d$  a GMM is defined by making each of the  $K$  components a Gaussian density with parameters  $\boldsymbol{\mu}_k$  and  $\Sigma_k$ . Each component is a multivariate Gaussian density

$$p(\mathbf{x}|\theta_k) = \frac{1}{(2\pi)^{d/2} |\Sigma_k|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\mathbf{x}-\boldsymbol{\mu}_k)}$$

with parameters  $\theta_k = \{\boldsymbol{\mu}_k, \Sigma_k\}$ . The parameters of the GMM are obtained by the Expectation-Maximization (EM) algorithm. The EM algorithm is an iterative algorithm that starts from some initial estimate of  $\Theta$  and then proceeds to iteratively update  $\Theta$  until convergence is detected. Each iteration consists of an E-step and an M-step. In the E-step,  $\omega_{ik}$  is computed following equation 3.1 for all data points  $\mathbf{x}_i$  and all mixture components  $k$ . For each data point  $\mathbf{x}_i$  the weights are defined such that  $\sum_{k=1}^K \omega_{ik} = 1$ . In the M-step, these membership weights and the data are used to calculate new parameter values

$$\alpha_k^{\text{new}} = N_k/N, \quad \boldsymbol{\mu}_k^{\text{new}} = (1/N_k) \sum_{i=1}^N \omega_{ik} \cdot \mathbf{x}_i$$

and  $\Sigma_k^{\text{new}} = (1/N_k) \sum_{i=1}^N \omega_{ik} \cdot (\mathbf{x}_i - \boldsymbol{\mu}_k^{\text{new}})(\mathbf{x}_i - \boldsymbol{\mu}_k^{\text{new}})^T,$

where  $N_k = \sum_{i=1}^N \omega_{ik}$  is the effective number of data points assigned to component  $k$ . After the new parameters are computed, the M-step is complete and the weights are recomputed in the E-Step. Then the parameters are again recomputed in the M-Step. Each pair of E and M-step is considered as one iteration.

For the second stage of the estimation process, the copula parameters are estimated. Extending multivariate Archimedian Copulas to different dependencies across link pairs is not straightforward as shown by [17], which would go beyond the scope of applying copulas in our use case for real world applications. Thus, an Archimedian Copula with single dependency parameter was fitted by using Maximum Likelihood Estimation. For  $d$ -variate observations  $\mathbf{x} = (x_{i1}, \dots, x_{id})^t$  with  $i \in \{1, \dots, n\}$  the log likelihood is given by

$$l(\boldsymbol{\theta}; \mathbf{x}) = \sum_{i=1}^n \log c(F_1(x_{i1}; \beta_1), \dots, F_d(x_{id}; \beta_d); \boldsymbol{\theta}),$$

where  $\boldsymbol{\theta}$  is the copula parameter space  $\beta_1, \dots, \beta_d$  are the marginal parameters.

### 3.2.2 Goodness of Fit Tests

For model testing and verification the following goodness of fit tests are used. Let  $x_1, x_2, \dots, x_n$  be an sample of  $n$  independent observations from a distribution with CDF  $F_N(x)$ . In order to test the null hypothesis  $H_0 : F_n(x) = F_H(x)$ , where  $F_n(x)$  and  $F_H(x)$  define the empirical CDF and the hypothetical CDF, respectively. In our case,  $F_H(x)$  is the historical path travel

time distribution and  $F_n(x)$  is the path travel time distribution estimated by each model, i.e. the various Copula Models and Convolution, respectively. The discrepancy between the two distributions can be either measured by

$$D = \max_x |F_n(x) - F_H(x)|$$

where  $D$  is referred to as the Kolmogorov-Smirnov statistic (KS) ([55]), or using quadratic statistics,

$$W^2 = \int_{\text{Ran } x} [F_n(x) - F_H(x)]^2 dF(x)$$

which is denoted as Cramer-von-Mises statistic (CVM) ([5]). Hence, the KS statistic gives us the largest discrepancy in a single point, while the CVM statistic describes the discrepancy over the total spectrum of the distribution.

### 3.2.3 Case Study: Urban Arterial

We evaluate the Copula Model by comparing it with the convolution method and an empirical distribution which we derived directly from historical data. For that purpose, we filtered the travel time samples for our study sites by Drive-ID obtaining actual through movements of all links of the paths for each vehicle. The samples with same Drive-ID corresponding to one vehicle were then added in order to derive the historical path travel time distribution. As the Drive-ID changes frequently during a trip due to data privacy, we can track a vehicle only for a limited amount of links. Thus, the length of our empirical path is restricted by this measure.

#### Study Site Description

This study site is comprised of ten segments with a total length of 586 m on Leopoldstraße, a major urban arterial in Munich, Germany. The speed limit on this arterial is 50 km/h. A schematic illustration can be found in Figure 3.1. The arterial consists of two signalized intersections. Furthermore, there is one bus lane, which stretches from Münchner Freiheit until Hohenzollernstraße, and there is signal control both at the start and at the end of the bus lane. The travel time data was collected over a period of one year, from 01. March 2013 until 01. March 2014. For the through movement of the arterial, i.e. trips of vehicles with constant Drive-ID, 4495 trips were recorded.



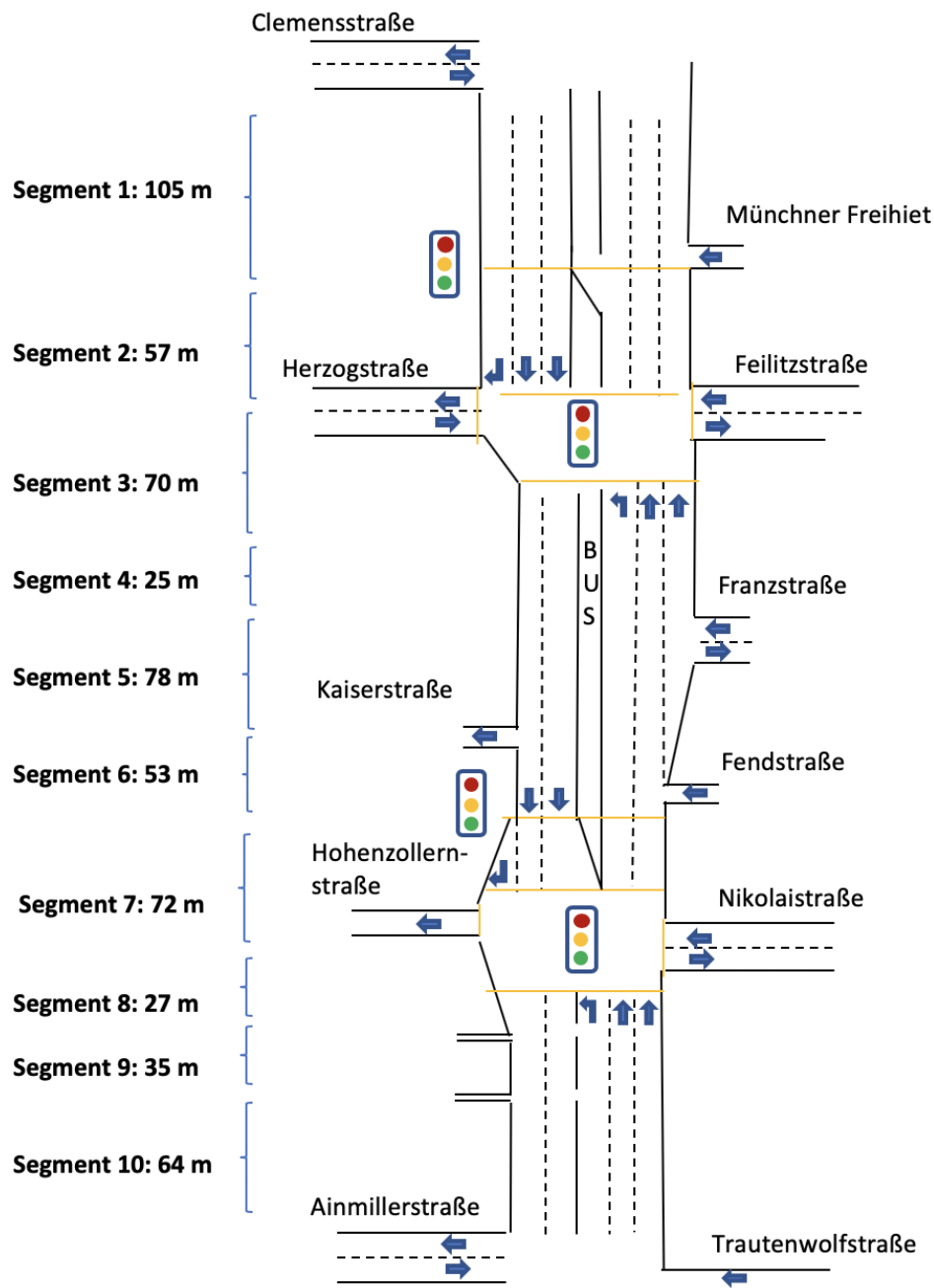


FIGURE 3.1: Schematic illustration of the urban arterial on Leopoldstraße in Munich.

### Link Travel Time Distributions and Correlation

In [13] the authors suggested that travel time on urban arterials follow multi-modal distribution with three peaks. They chose a GMM with three components GMM to estimate marginal distribution. Also for this study, GMM with three components showed an accurate fit. Figure ?? and 3.3 show the travel time distributions and the GMM estimation for all ten links of the study site. The parameters of the GMM can be found in Table 3.1. The first component of GMM describes the free flow state. The second component implies partially delayed vehicles, while the third component reflects the congested situation. For the longest segment, Segment 1 in Figure 3.9a, with a length of 105m, three peaks can be observed. For the segments with a shorter length, the observable distinction is not that obvious, however GMM with three components still shows an accurate fit.

TABLE 3.1: GMM Parameters for urban segment travel time distribution estimation.

Segment	Mean ( $\mu$ )	Sigma ( $\sigma$ )	Weight ( $\pi$ )
Segment 1	(16.08, 31.41, 62.92)	(5.25, 9.79, 12.65)	(0.31, 0.34, 0.34)
Segment 2	(5.41, 8.86, 16.31)	(1.44, 2.68, 5.58)	(0.52, 0.38, 0.09)
Segment 3	(8.92, 14.55, 29.37)	(2.03, 4.06, 9.55)	(0.43, 0.34, 0.22)
Segment 4	(3.11, 5.72, 10.33)	(0.72, 1.69, 3.26)	(0.43, 0.38, 0.17)
Segment 5	(9.46, 17.58, 36.01)	(2.18, 5.26, 10.38)	(0.33, 0.38, 0.28)
Segment 6	( 6.43, 12.26, 29.55)	(1.53, 3.94, 5.36)	(0.46, 0.35, 0.17)
Segment 7	(8.24, 13.30, 27.82)	(1.68, 3.75, 10.94)	(0.54, 0.37, 0.07)
Segment 8	(2.76, 3.97, 7.09)	(0.48, 0.91, 2.69)	(0.48, 0.38, 0.12)
Segment 9	(3.59, 5.42, 10.17)	(0.64, 1.34, 3.94)	(0.52, 0.35, 0.11)
Segment 10	(6.67, 11.29, 22.46)	(1.32, 3.22, 8.64)	(0.52, 0.35, 0.11)

In order to investigate segment correlation, scatter diagrams for successive segments are shown in Figure 3.4. In addition, Kendall's tau for each segment pair is given. The values range from 0.318 to 0.835 showing that segment correlation does exist. The mean value for Kendall' tau for neighbouring links is  $\bar{\tau}_{\text{urban}} = 0.58$ . Dependency is a function of network connectivity which is shown in [7] and [8]. In our study site the segments are connected and vehicles are able to travel all segments without the need to leave the route. However, as there are several factors affecting the travel time of each segment and the correlation in between segments, such as signal control, turning lane, parking

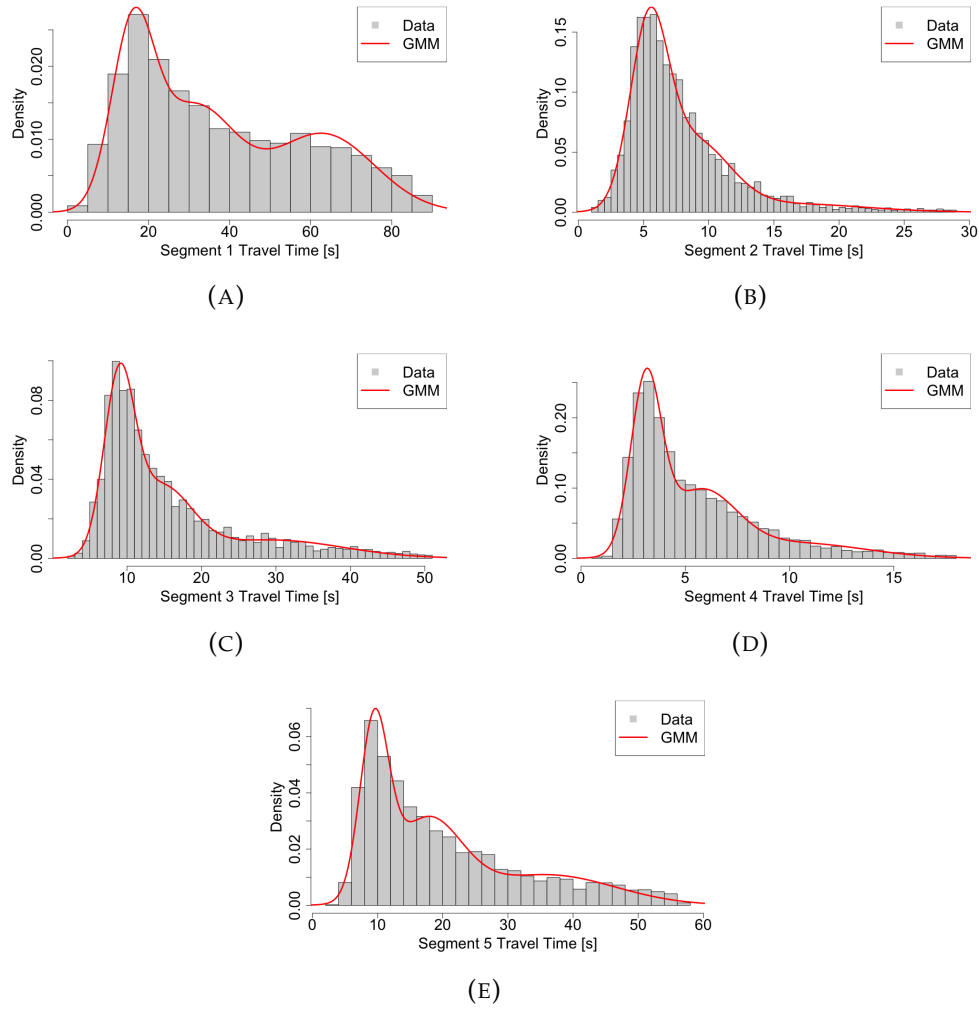


FIGURE 3.2: Urban Segment travel time distributions: a) Segment 1, b) Segment 2, ..., e) Segment 5

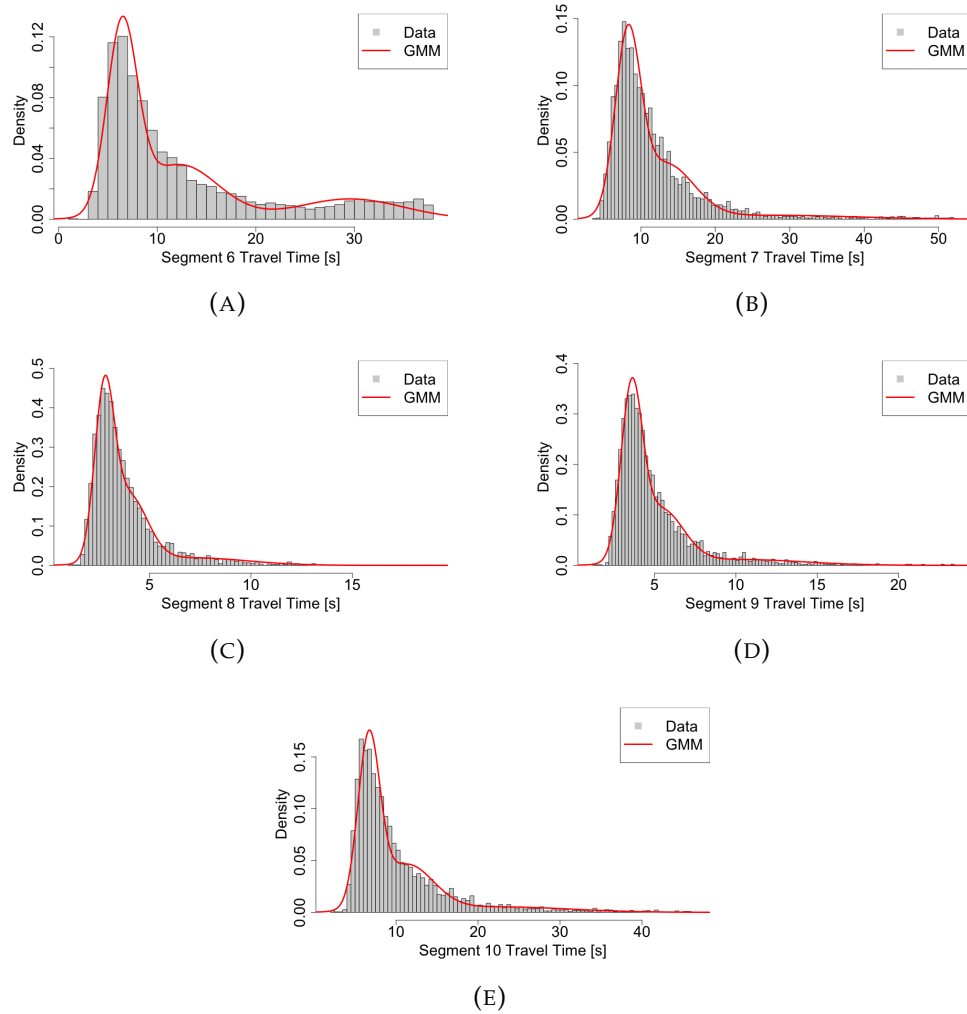


FIGURE 3.3: Urban Segment travel time distributions: a) Segment 6, b) Segment 7, ..., e) Segment 10

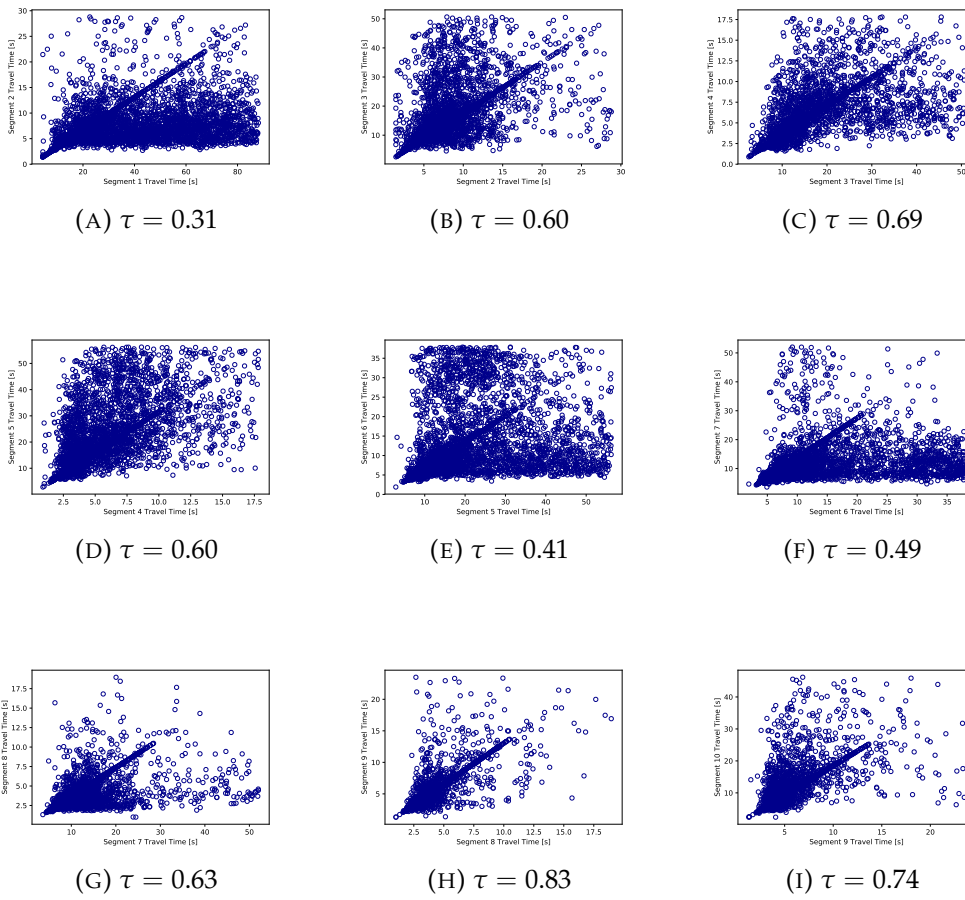


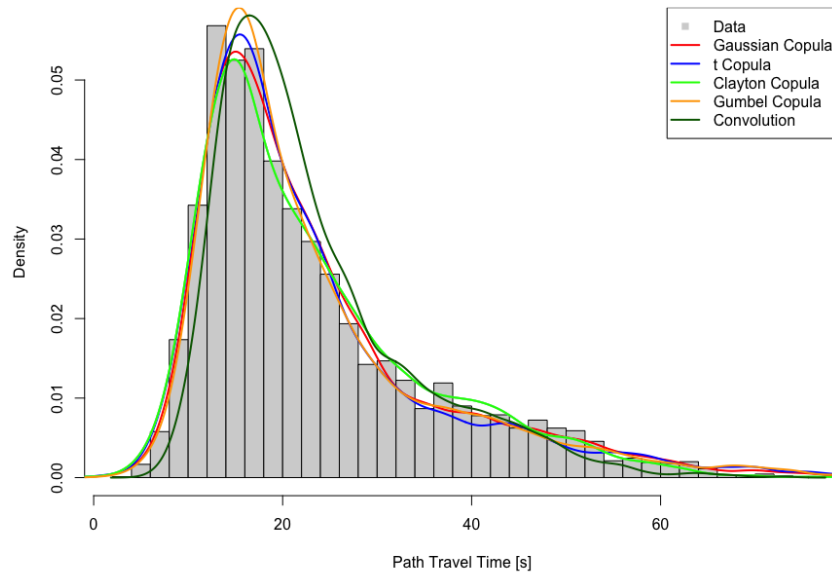
FIGURE 3.4: Scatter Diagrams for Urban Segment travel time distributions: a) Segment 1 and Segment 2 , b) Segment 2 and Segment 3 , ..., i) Segment 9 and Segment 10.

lane, weather, and congestion, the correlation structure is complex and differs for each segment pair. For each segment pair the values of individual travel times scatter in a wide range of the joint distribution space. Only segment pairs shown in 3.4g, 3.4h, and 3.4i, which are relatively short and do not have signal control nor turning lanes show a strong lower tail dependence. This is also resembled by a high value for Kendall's tau, with 0.639, 0.835, and 0.748, respectively. The line through the origin is caused by the estimation of the velocities of probe vehicle described in the Data chapter. If one GPS point is located before Segment  $i$  and the next GPS point is located behind Segment  $j$ , the velocity for both segments is equal. Therefore, travel time of Segment  $i$  and Segment  $j$  are proportional, which causes the line through origin, where the corresponding samples lie.

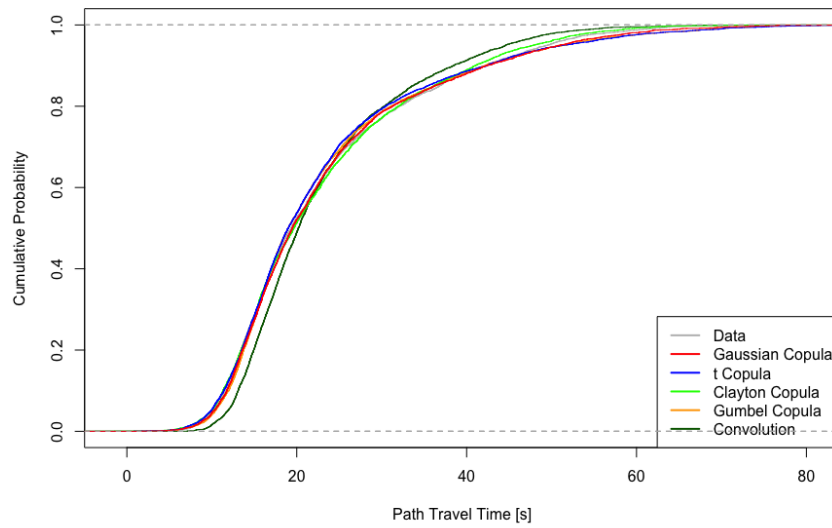
### Estimation of Path Travel Time Distribution

Now that we have described our link travel times by continuous distributions, we can proceed to the second stage and fit a Copula Model for estimating path travel time distribution for the total arterial. In order to assess the scalability of the Copula Model in terms of number of links, we first estimate the distribution for a path comprised of the first two segments, which we will refer to as *2D Copula Model*. Then, we estimate the distribution for the total arterial, which we will refer to as *10D Copula Model*. We compare the following copulas: Gauss, Student-t, Clayton and Gumbel. For each of these copulas we fit a corresponding Copula Model. We start with the 2D Copula Models. Figure 3.5 shows the PDF and CDF for path travel time distribution by the respective 2D Copula Models, convolution and empirical data. In order to evaluate the performance of Copula Model and convolution, we compare the distributions obtained by the respective models with the empirical one. The goodness of fit tests can be found in Table 3.2. As both KS and CVM describe deviations from the reference distribution, a lower value for both statistics indicates a better fit. We can observe, that each Copula Model performs better than the convolution. That is due to their ability to incorporate segment correlation, while the convolution assumes independence. The 2D Clayton Copula Model has the lowest values for both KS and CVM and, thus, is the most accurate model. A possible reason may be its ability to capture lower tail dependence, which we can partially observe in the scatter diagrams in Figure 3.4.

Now we increase the number of segments. PDF and CDF for path distribution estimation by the 10D models is shown in Figure 3.6. The corresponding



(A)



(B)

FIGURE 3.5: Path Distribution estimation by 2D Models: PDF (A) and CDF (B).

TABLE 3.2: Goodness of fit tests of the 2D models for path TTD estimation.

2D Model	KS	CVM
Convolution	0.031	0.023
Gauss	0.015	0.004
Student-t	0.016	0.010
<b>Clayton</b>	<b>0.014</b>	<b>0.003</b>
Gumbel	0.024	0.008

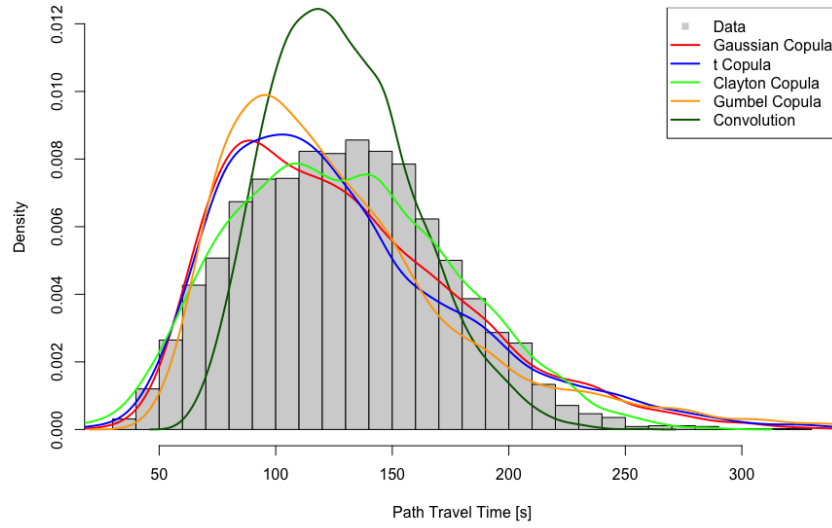
goodness of fit tests are listed in table 3.3. Compared to the results for the 2D models, the inaccurate estimation of the convolution as well as the superior estimation of the Copula Models is more distinct. Again, each Copula Model performs better than the convolution, while the Clayton copula shows the best fit.

TABLE 3.3: Goodness of fit tests and parameters of the 10D models for path TTD estimation.

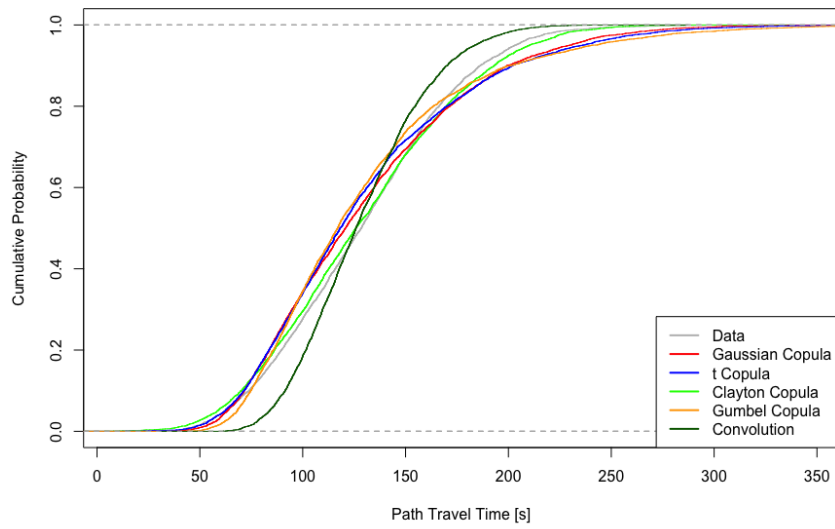
10D Model	KS	CVM
Convolution	0.113	0.760
Gauss.	0.046	0.340
Student-t	0.053	0.472
<b>Clayton</b>	<b>0.026</b>	<b>0.061</b>
Gumbel	0.054	0.579

In order to test the scalability of the Copula Model, we iteratively increased the number of segments from two to ten. To show the results for every number of segments would be redundant, that is why we only show the results for the 2D and 10D Copula Model. But what we do include here, is an overview of the goodness of fit for every iteration. As CVM describes the deviation of the estimation to the reference distribution over the total spectrum, while KS only gives the highest deviation of one point, we choose to monitor CVM. Furthermore, we only compare the convolution to the best fitting copula, i.e. Clayton. Figure 3.7 shows CVM for each iteration. The accuracy of the convolution decreases with the number of segments, while the accuracy of the Clayton copula stays nearly constant.





(A)



(B)

FIGURE 3.6: Path Distribution estimation by 10D Models: PDF (A) and CDF (B).

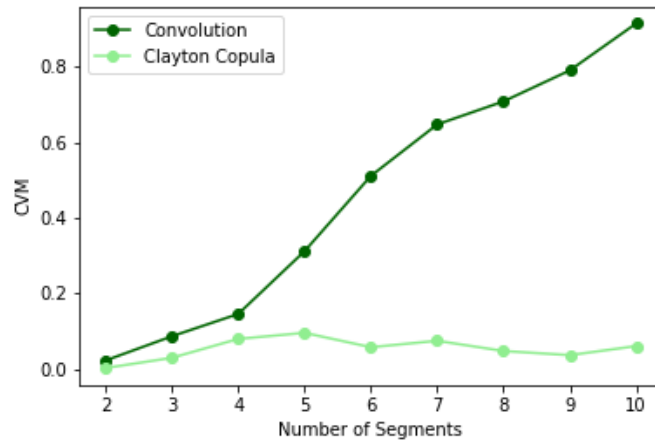


FIGURE 3.7: CVM Overview

### 3.2.4 Case Study: Freeway Arterial

#### Study Site Description

This study site is comprised of ten segments of the freeway facility A9 north of Munich, Germany, with a total length of 1164 m. It is illustrated in Figure 3.8. There are three through lanes with occasional on ramp lanes. The speed limit is only valid for certain time of the day and traffic conditions. Depending on the traffic situation the speed limit takes values of 80 km/h, 100 km/h or 120 km/h. Otherwise there is no speed limit. The recommended velocity on German freeways in that case is 130 km/h .

#### Link Travel Time Distributions and Correlation

In [61] the authors suggested that freeway travel time consist of four phases. The first phase corresponds to a free flow state, where median travel times are low and the spreads of the distribution are small. The distribution is approximately symmetric. The second phase describes the congestion onset, where median travel times are still low. The third phase is congested traffic, where median travel times are high, while the distribution is wide. In these periods the congestion can be expected in different degrees of severity, resulting in a wide range of possible travel times. The fourth state describes congestion dissolve, where median travel times are low, but the distribution is skewed to the left. This resembles that in most cases congestion has dissolved but in a decreasing number of cases still congestion occurred. Figure 3.9 and 3.10 show the distributions of travel times for each segment of the freeway arterial. For estimation of an underlying continuous distribution, a GMM with four

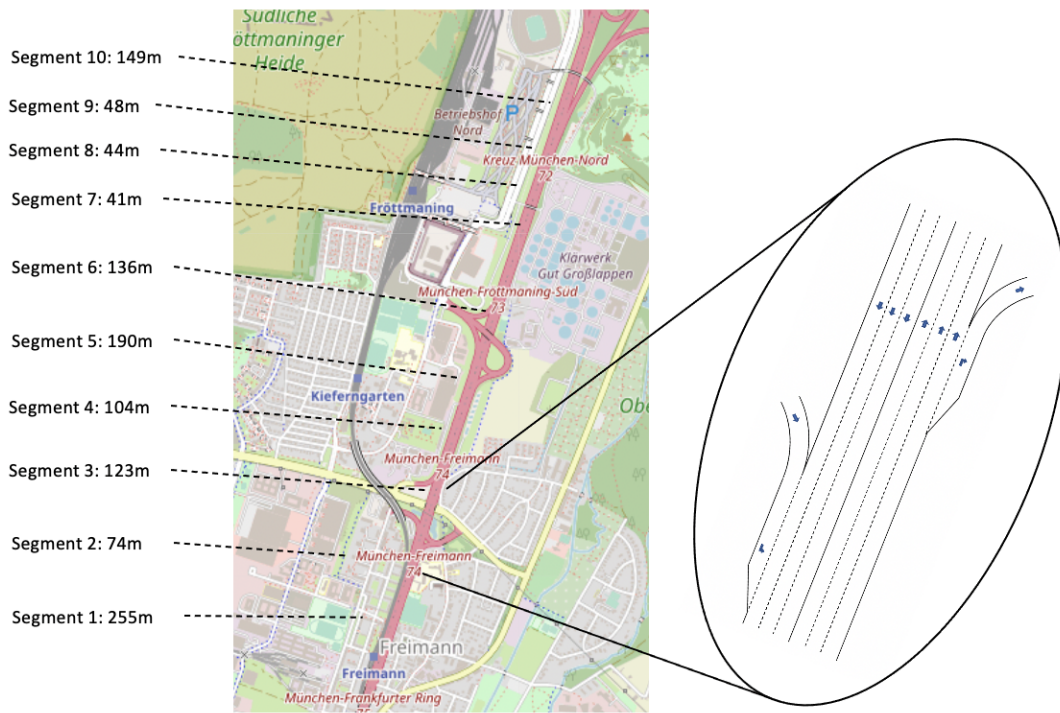


FIGURE 3.8: Schematic illustration of the freeway arterial on A9.

components was found as best fitting. The parameters of the GMM can be found in Table 3.4. In accordance with the results of [61] the four components can be interpreted as follows. The first GMM component reflects the free flow phase. The second component resembles the phase of congestion onset, as well as the congestion dissolve. The third and fourth components correspond to states with congested traffic with different levels of congestion, respectively.

Figure 3.11 shows the scatter diagrams for each successive link pair. A clear left tail dependence structure can be observed for every link pair with values of Kendall's tau ranging from 0.62 to 0.85 with a mean value of  $\bar{\tau}_{\text{freeway}} = 0.73$ . Compared to the urban arterial, where  $\bar{\tau}_{\text{urban}} = 0.58$ , the correlation is stronger and its structure is more distinct. That is due to the mostly uninterrupted traffic flow on the freeway compared to the interrupted traffic flow on the urban arterial.

### Estimation of Path Travel Time Distribution

Analogously to the case study on the urban arterial, we compare Copula Models of the same copulas, i.e. Gauss, Student-t, Clayton and Gumbel, with convolution and the empirical distribution. We also start by creating 2D models for the first two segments and compare the 10D models for the total

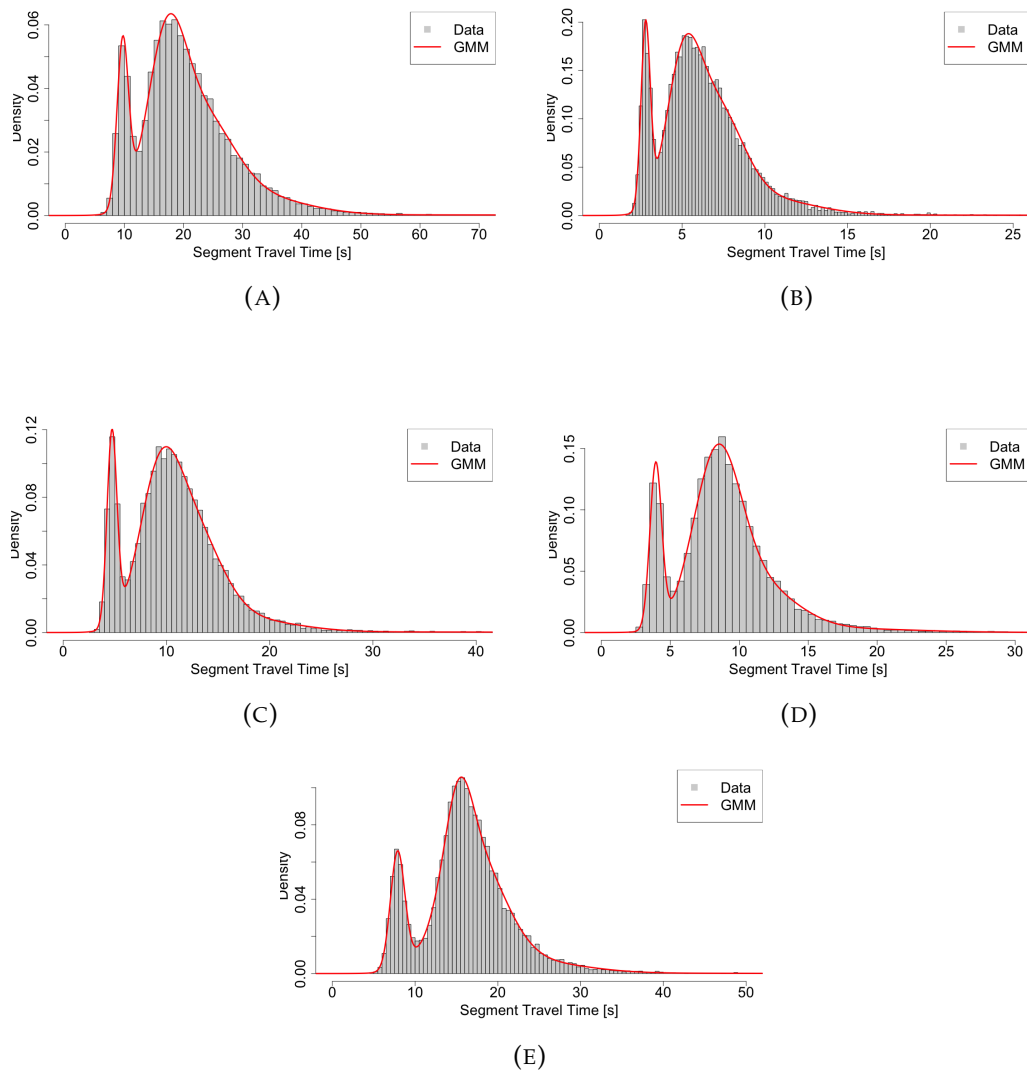


FIGURE 3.9: Freeway Segment travel time distributions: a) Segment 1 , b) Segment 2 , ..., e) Segment 5

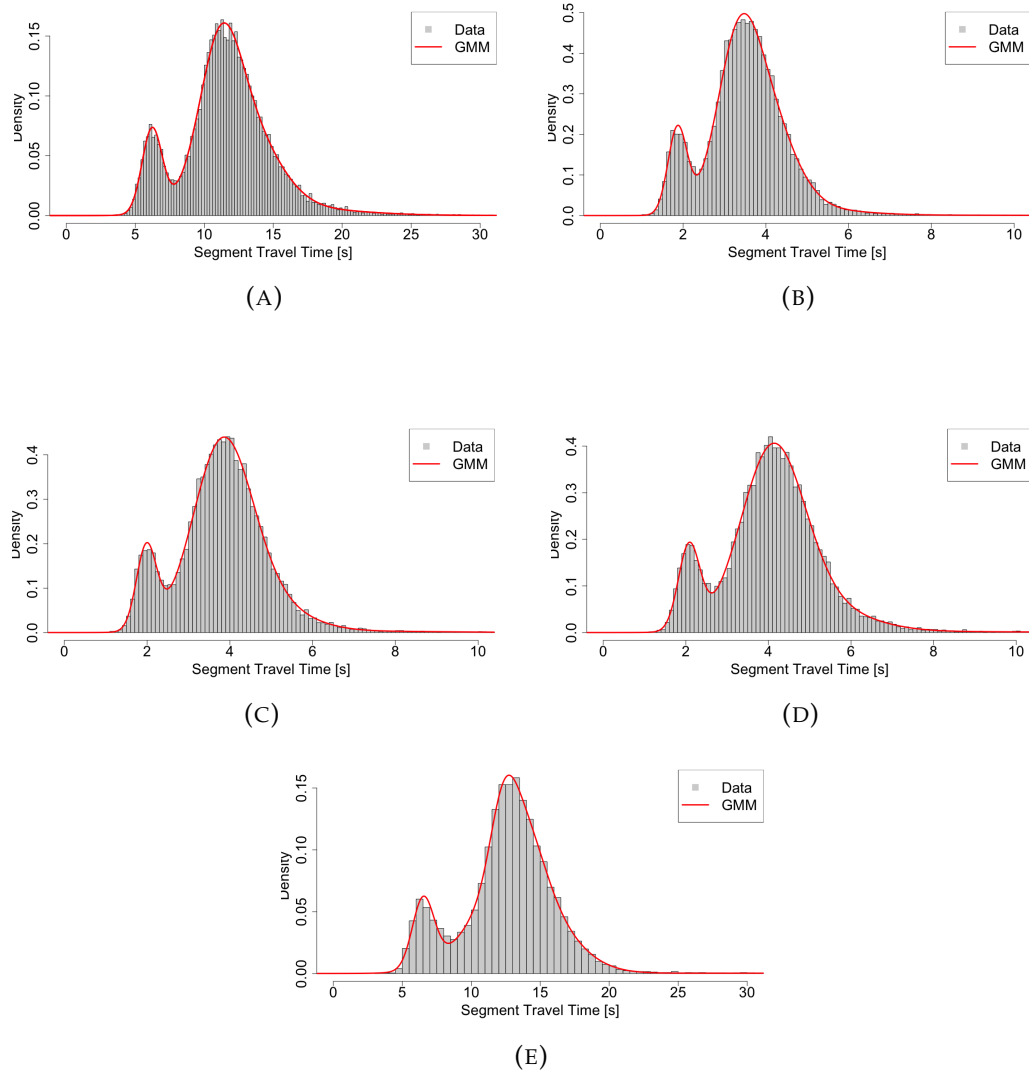


FIGURE 3.10: Freeway Segment travel time distributions: a) Segment 6 , b) Segment 7 , ..., e) Segment 10

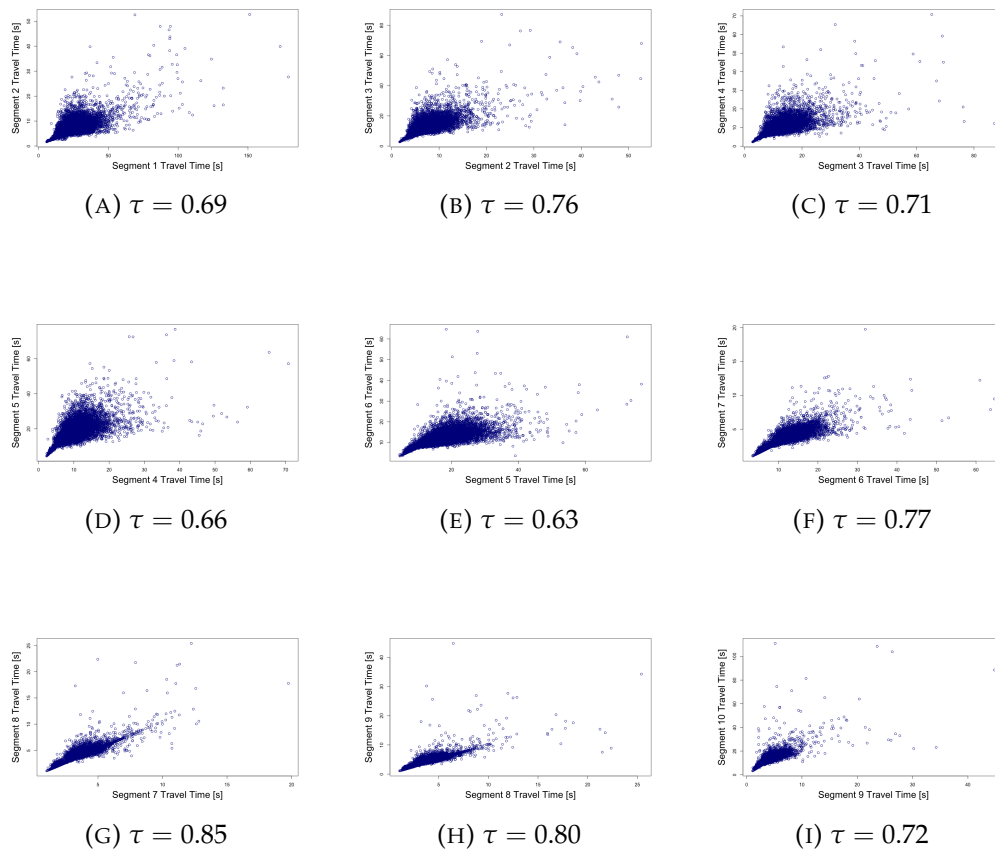


FIGURE 3.11: Scatter Diagrams for freeway Segment travel time distributions: a) Segment 1 and Segment 2 , b) Segment 2 and Segment 3 , ..., i) Segment 9 and Segment 10.

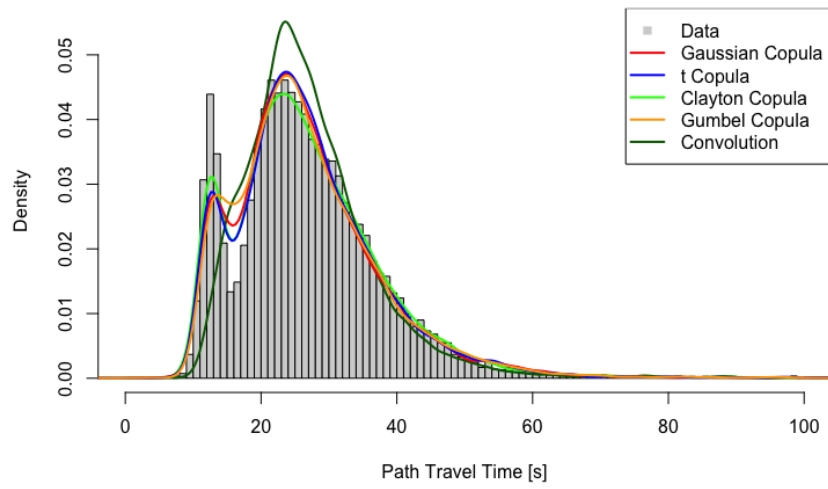
TABLE 3.4: GMM Parameters for highway segment travel time distribution estimation.

Seg.	Mean ( $\mu$ )	Sigma ( $\sigma$ )	Weight ( $\pi$ )
1	(9.70, 17.06, 23.06, 32.16)	(0.96, 3.24, 5.28, 8.46)	(0.12, 0.37, 0.36, 0.11)
2	(2.79, 5.03, 7.00, 10.16)	(0.27, 1.01, 1.60, 2.84)	(0.12, 0.32, 0.41, 0.11)
3	(4.72, 9.22, 12.32, 17.70)	(0.47, 1.97, 2.80, 4.80)	(0.13, 0.35, 0.40, 0.08)
4	(3.94, 8.36, 11.25, 17.40)	(0.42, 1.71, 2.77, 5.30)	(0.14, 0.56, 0.24, 0.04)
5	(7.88, 15.23, 16.96, 24.10)	(0.86, 1.65, 3.71, 6.27)	(0.13, 0.19, 0.56, 0.09)
6	(6.21, 11.13, 12.51, 16.92)	(0.72, 1.44, 2.49, 4.40)	(0.12, 0.32, 0.48, 0.05)
7	(1.86, 3.30, 3.38, 4.95)	(0.23, 0.50, 0.71, 1.23)	(0.12, 0.35, 0.46, 0.04)
8	(1.98, 3.81, 4.65, 7.15)	(0.24, 0.71, 1.04, 2.09)	(0.11, 0.69, 0.17, 0.01)
9	(2.07, 4.09, 5.12, 8.05)	(0.25, 0.77, 1.20, 2.40)	(0.11, 0.71, 0.15, 0.01)
10	(6.51, 13.25, 13.62, 12.31)	(0.78, 3.06, 1.48, 0.94)	(0.10, 0.18, 0.11, 0.57)

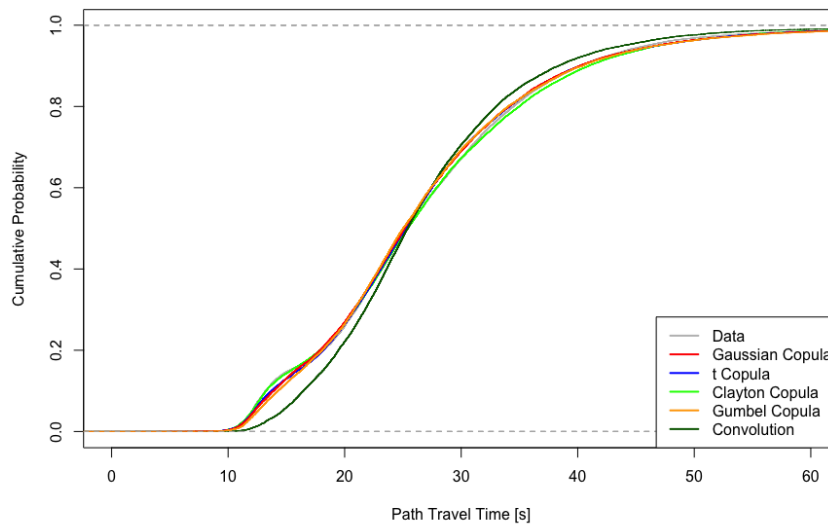
path afterwards. The PDF and CDF for the 2D models and 10D models are shown in Figure 3.12 and Figure 3.13, respectively. The goodness of fit tests for the 2D models can be found in Table 3.5 and the tests for the 10D models are listed in Table 3.6. We can observe analogous results to the ones in the urban case. For both 2D and 10D models every Copula Model performs better than the convolution, while the Clayton copula shows the best fit. That is, because we have a clear left tail dependence structure for the neighbouring segments, and the Clayton copula is able to capture that specific structure. In Figure 3.14 we compare CVM of the convolution with CVM of the Clayton copula for path travel time estimation reaching from two segments until ten segments. Analogously to the urban case study, we can observe a decreasing accuracy of the convolution with the number of segments compared to the Copula Model.

TABLE 3.5: Goodness of fit tests of the 2D models for highway path TTD estimation.

2D Model	KS	CVM
Convolution	0.089	0.056
Gauss	0.023	0.004
Student-t	0.018	0.003
<b>Clayton</b>	<b>0.011</b>	<b>0.001</b>
Gumbel	0.033	0.007



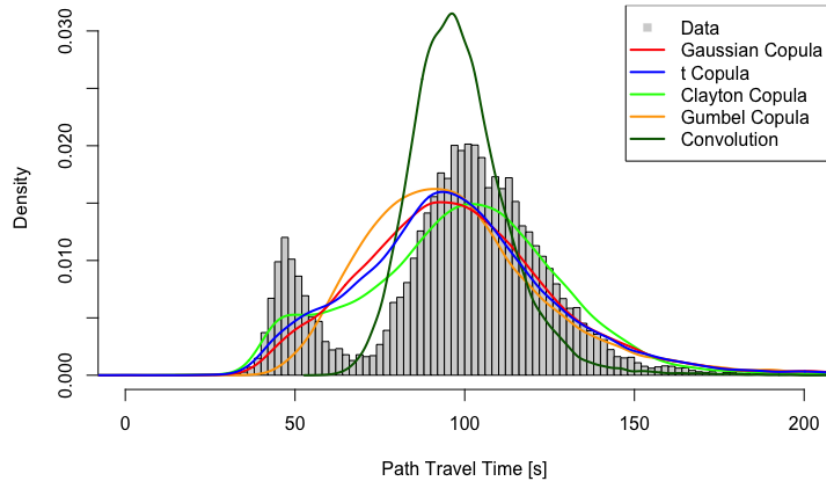
(A)



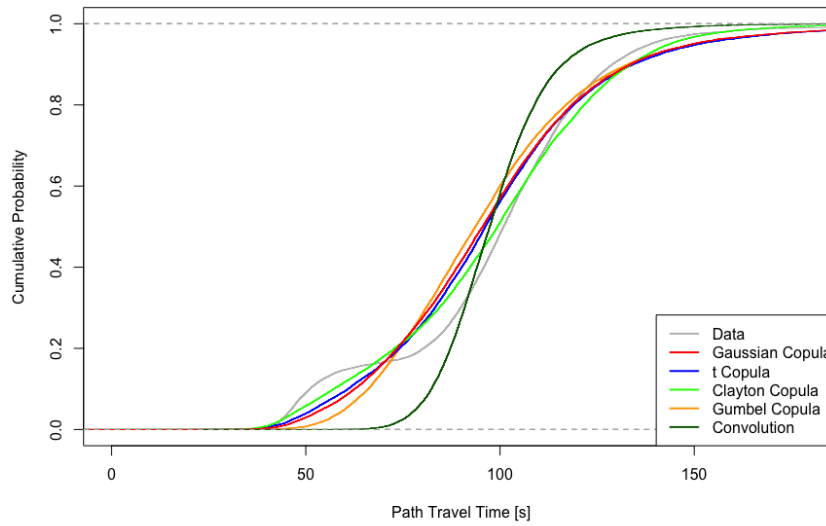
(B)

FIGURE 3.12: Path Distribution estimation by 2D Models: PDF (A) and CDF (B).





(A)



(B)

FIGURE 3.13: Path Distribution estimation by 2D Models: PDF (A) and CDF (B).

TABLE 3.6: Goodness of fit tests of the 10D models for highway path TTD estimation.

10D Model	KS	CVM
Convolution	0.159	1.244
Gauss	0.073	0.386
Student-t	0.060	0.294
<b>Clayton</b>	<b>0.040</b>	<b>0.137</b>
Gumbel	0.060	0.294

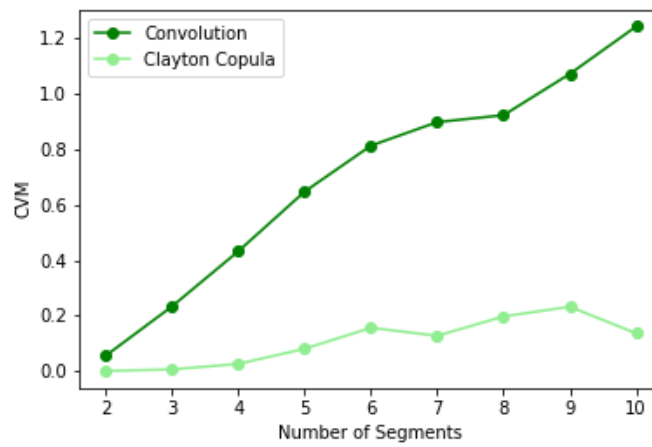


FIGURE 3.14: CVM Highway Overview

### 3.3 DDC Model

We have showed that copulas provide the ability for modelling path travel time distribution from link distributions as they are able to incorporate the dependence structure between successive links. Now, we can proceed to the next step towards developing a stochastic router. For that purpose we have to ask the following question. Can we use the Copula Model in real world application? The answer is no, simply because its architecture is too complex. Marginal distribution have to be estimated, a multivariate distribution has to be generated and samples need to be generated from that distribution. Due to these procedure stages, the Copula Model may be limited in its applicability due to long computational time. Thus, we need to find another way to use copulas for incorporating correlation into the modelling of travel time distribution.

This work proposes a novel approach for modelling travel time distribution by using *copula-based dependent discrete convolution (DDC)*. In [70] DDC was previously introduced for power system uncertainty analysis. The results indicated that DDC improves the calculation accuracy compared with traditional convolution by successfully capturing the dependency of wind power. The theory of DDC was already elucidated in Chapter 1. Compared to the Copula Model this approach allows for a simplified and fast computation of the travel time distribution as shown in Figure 3.15.

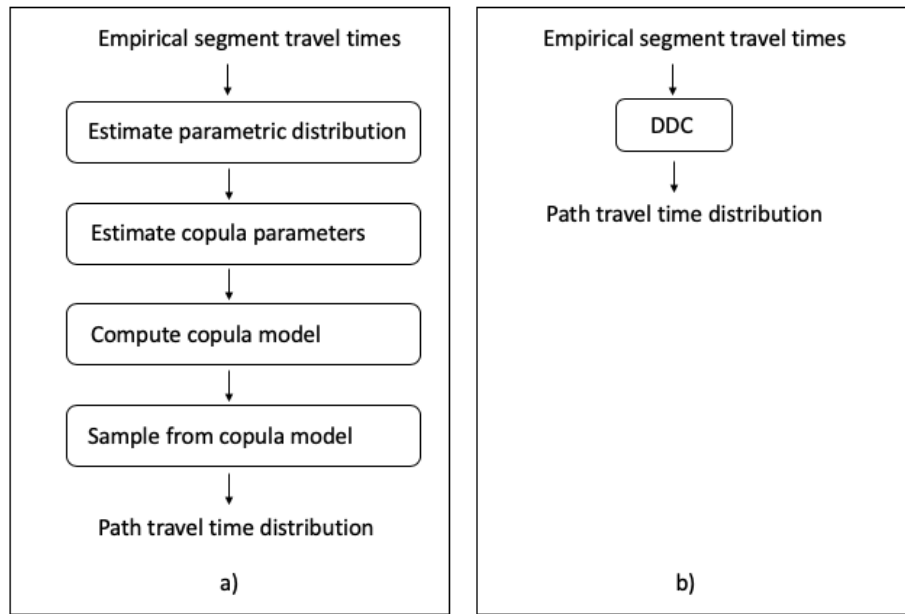


FIGURE 3.15: Comparison of the Copula Model (a) and the DDC (b) for estimating path travel time distribution from empirical segment travel times.

### 3.3.1 Methodology

DDC requires a distinct factor describing correlation between two random variables for each iteration. Therefore, we cannot apply DDC for more than two variables at the same time as we can when using convolution. For estimating path travel time distribution, where we need to aggregate multiple link distributions, we propose the *DDC Model*. Here, we apply the DDC pairwise in an upside down pyramid scheme on all link distribution until path distribution is reached as illustrated in Figure 3.16. For comparison with the Copula Model we will apply the DDC Model on the same study sites presented above, which consist of ten links, respectively. In Iteration 1 we apply DDC on each link pair and obtain travel time distributions for "1-2",

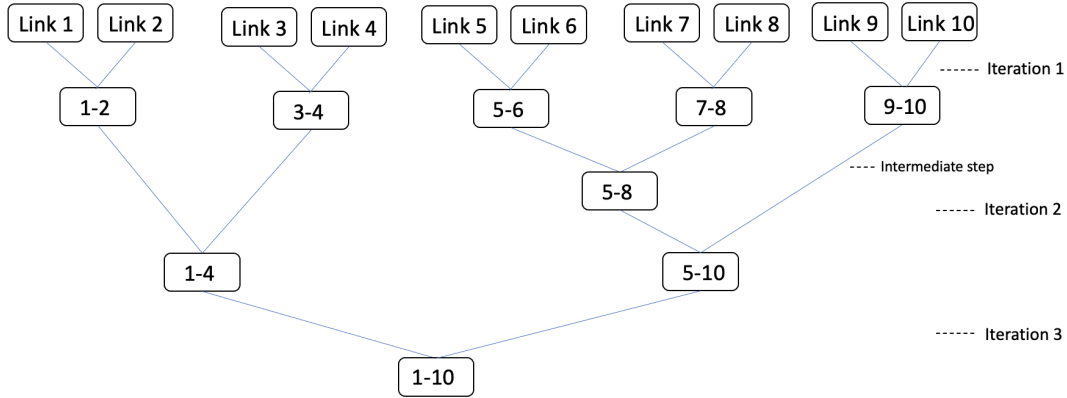


FIGURE 3.16: Illustration of the DDC Model for modelling path travel time distribution for ten link distributions. The path distribution consisting of Link 1 and Link 2 is denoted as "1-2", and so on.

..., "9-10". In Iteration 2 the number of DDC inputs is not an integer multiple of two. For that reason, we introduce an intermediate step within Iteration 2. We first apply DDC on the first two pairs to obtain "1-4" and "5-8". In the intermediate step we then apply DDC on "5-8" and "9-10". Eventually, we aggregate "1-4" and "5-10" in Iteration 3 to obtain the total path distribution "1-10".

### 3.3.2 Comparison to Copula Model

We estimate path travel time distribution for the same two study sites as above, i.e. urban arterial and freeway arterial, using DDC Model and Copula Model in order to compare both methods. As the Clayton copula showed the best results in the previous case studies, we will use it in the DDC. We compare it with the Clayton Copula Model. We will directly show the path distribution for the total arterial of ten segments.

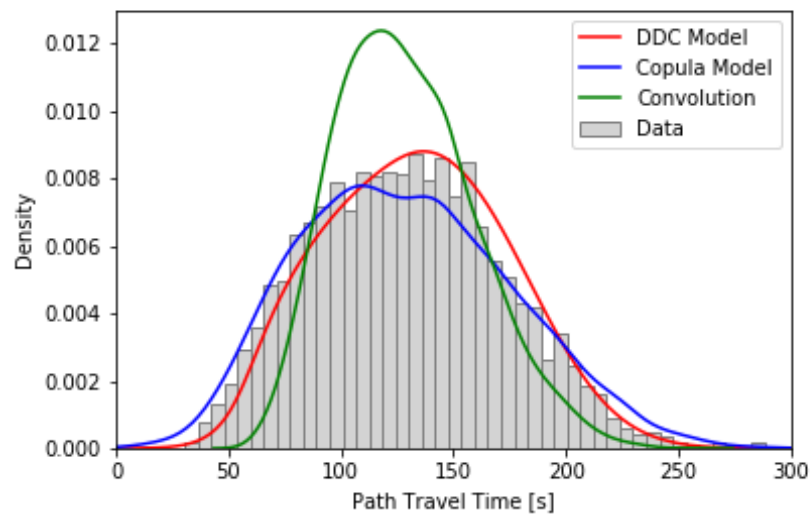
The urban arterial as well as the freeway arterial and their corresponding travel time data have already been described in Subsection 3.2.3 and Subsection 3.2.4, respectively. In order to determine the remaining correlation needed for the DDC Model, we compute Kendall's tau for the respective link pairs. For Iteration 1 we need to determine the correlation between neighbouring links, which we have already shown in the previous section. Next, we add the empirical travel times for the individual links in order to obtain travel times for "1-2", ..., "9-10". We proceed analogously for obtaining historical travel times for "1-4", "5-8", and "5-10". Then we can calculate the respective values

for Kendall's tau, which we need in each Iteration step. These are shown for both study sites in Table 3.7.

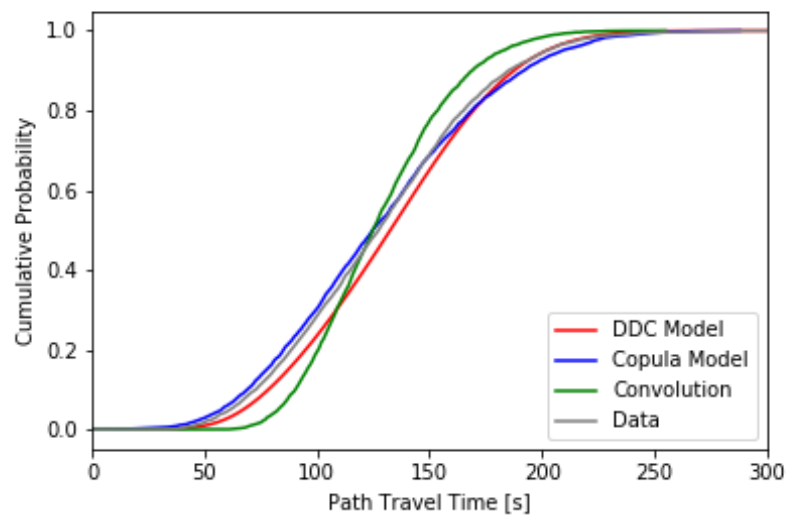
TABLE 3.7: Values for Kendall's tau for each Iteration step of the DDC Model for urban and freeway arterial.

	<b>Urban</b>	<b>Freeway</b>
Link Pair	$\tau(\cdot, \cdot)$	$\tau(\cdot, \cdot)$
Link 1, Link 2	0.32	0.69
Link 3, Link 4	0.70	0.72
Link 5, Link 6	0.42	0.63
Link 7, Link 8	0.64	0.85
Link 9, Link 10	0.75	0.72
1-2, 3-4	0.21	0.57
5-6, 7-8	0.30	0.59
5-8, 9-10	0.23	0.52
1-4, 5-10	0.24	0.40

Now, we can apply the DDC Model for estimating path travel time distribution. Figure 3.17 and Figure 3.18 show the PDFs and CDFs for path travel time distribution estimation obtained by each model for the urban arterial and freeway arterial, respectively. The corresponding goodness of fit tests are listed in Table 3.8. Goodness of fit tests show that the Copula Model estimates path travel time distribution slightly more accurately than the DDC, while the path travel time distribution estimate of the DDC is still more accurate than convolution by far. However, when comparing computing time, the power of the DDC Model is unveiled. Figure 3.19 shows the computing time per number of segments. The computing time is illustrated for the iterative aggregation of the total path comprised of ten segments on a machine with a 2.8 GHz core and 16GB memory. Furthermore, a fictional path is computed using both DDC and Copula Model by repeatedly aggregating the ten segments in order to estimate the scaling of computing time. For application in route guidance systems, a path consisting of 100 segments is realistic. While the DDC takes approximately 3 seconds, which is totally feasible for applications, the Copula Model takes approximately 20 seconds, which is not eligible for real world applications.

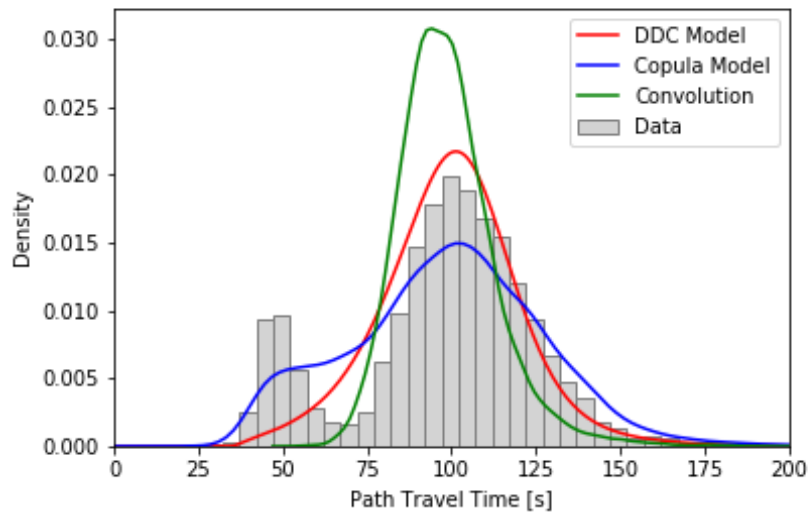


(A)

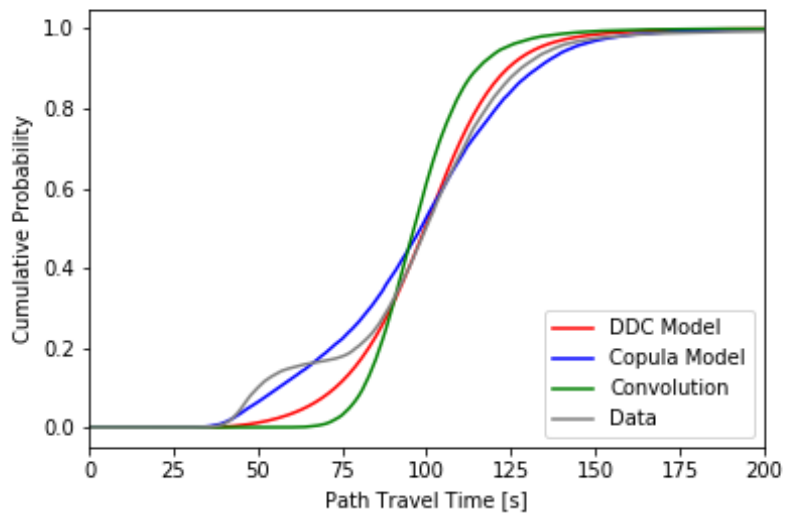


(B)

FIGURE 3.17: Path Distribution estimation for each model for the urban arterial: PDF (A) and CDF (B).



(A)



(B)

FIGURE 3.18: Path Distribution estimation for each model for the freeway arterial: PDF (A) and CDF (B).

TABLE 3.8: Goodness of fit tests of the for Path travel time distribution estimated by the respective models for the urban arterial.

Model	Urban		Freeway	
	KS	CVM	KS	CVM
DDC Model	0.054	0.196	0.115	0.336
Copula Model	0.026	0.061	0.040	0.137
Convolution	0.113	0.760	0.159	1.244

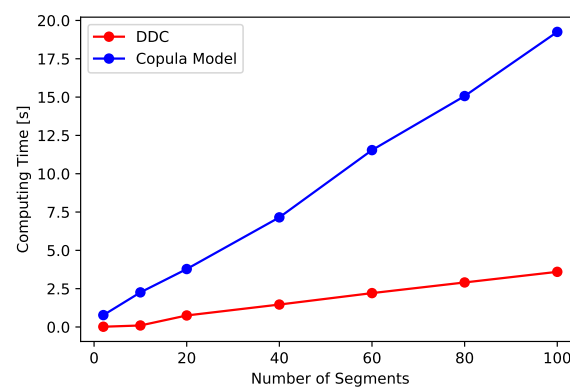


FIGURE 3.19: Computing time for the DDC and Copula Model



## Chapter 4

# Stochastic Routing

In this chapter we will use the methodology we have achieved so far and develop it further to a stochastic router. We have briefly mentioned the key idea of stochastic routing in the Introduction. Here, we go further into detail and give an overview of related work. Then, we elucidate the methodology behind our concept. Eventually, a case study is presented for evaluation.

### 4.1 Related Work

Finding a solution to the SSP is of high interest for both industry and academy as it is directly relevant for people's daily life ([11]). While in the traditional shortest path problem there are static discrete link costs, the SSP problem deals with dynamic probabilistic link costs. In shortest path problem, the definition optimality is straightforward as the optimal path is the one with the minimal travel time. In SSP problem, however, there is no unique definition of optimality resulting from uncertain user demands and traffic conditions ([53]). Hence, several optimality definitions can be found in the literature. One common approach is to set the least expected travel time as a criteria ([26], [42], [62], [66]). Optimal paths are then obtained by setting each link cost to its expected value and solving the corresponding deterministic problem. Another possibility is to take a minimal weighted combination of standard deviation and expected value for the travel time ([44], [45]). However, both of these approaches do not provide an opportunity to include user preferences into their optimality definition, e.g probability of arriving on time with respect to a deadline. For that purpose, it is necessary to work with probability distributions describing link travel times instead of deriving discrete values.

The optimal path is then defined as the path that maximizes the probability of arriving on time. In [24], instead of giving an a priori solution to the SSP problem, the aim is to identify which node should be visited next in order

to maximize the probability of on time arrival. For the formulation of the mathematical problem Bellman's principle of optimality is applied. The Picard method of successive approximations is used to estimate the unknown functions describing the maximum probability of arriving on time. As convolution is used in the successive approximation procedure, link travel times are treated as independent. In [25] a similar procedure is applied with the difference that for each link two possible states are assumed, i.e. one congested state and one state without congestion. The correlation between the states of adjacent nodes are considered by introducing two conditional probabilities. The authors stated that they proposed only two possible levels of link states for convenience, but this assumption is rather unrealistic as links may show different levels of congestion. In [45] an algorithm based on quasi-convex maximization is proposed to find the path with maximum probability of not exceeding a certain cost. However, this approach is limited by the assumption that link travel times follow a normal distribution and are independent. In [36] a stochastic motion planning algorithm and its application to traffic navigation is presented. The algorithm can determine a path for a origin – destination pair that maximizes the probability of arriving on time. Yet, the limitations of this procedure are also the assumption of a normal link travel time distribution and independence between link travel times.

In order to tackle these shortcomings of previous work the authors in [12] propose a formulation of the SSP problem as a cardinality minimization problem. This approach is directly based on travel time samples of each road link, which are obtained by probe vehicles. An  $l_1$ -norm minimization technique is applied to solve the cardinality problem. It is then reformulated as linear programming problem, which can be solved using state-of-the-art solvers. The authors claim that their approach solves the limitation of previous work. First, they claim that their method does not need any assumption on travel time distribution as it can work directly with historical travel time data. However, their problem formulation creates a different limitation as there has to be the same amount of travel time samples for each road link. In reality, this may not be the case. Hence, there are two possibilities for applying their method. Either, one compromises on a number of samples which is given by the link with the lowest amount of data, losing valuable data for other links, or, one uses methods to generate samples for the links with a low amount of data, where one needs to assume a underlying travel time distribution again. None of these choices is optimal. Secondly, the authors claim that their

approach does not rely on the assumption that links are independent of each other and can be applied to correlated links as well. However, the authors do not perform any correlation analysis on real data. For the evaluation they only use distributions for each link with random correlations. This may not reflect the true dependence structure of link travel times. Furthermore, the method is very expensive as it requires the enumeration of every possible path for one origin – destination pair.

In this thesis we propose a method for solving the SSP problem in terms of stochastic vehicle routing, which tackles the shortcomings of related state-of-the-art work. Firstly, our method does not require the assumption of an underlying distribution for link travel times. It can handle historical data directly without the strong restriction of needing the same amount of samples per link. Secondly, our method incorporates the true dependence structure between link travel times based on correlation analysis of real data.

## 4.2 Preliminaries

We first give a brief overview of the notation which is used in the upcoming methodology. We describe the road network as a directed graph  $G(N, L, P)$ , where  $N$  is the set of nodes  $N(|N| = n)$ ,  $L(|L| = m)$  is the set of links and  $P$  is the probability distribution of link travel times. A path connecting origin  $o \in N$  and destination node  $d \in N$  is denoted as  $k^{od}$  and its corresponding travel time is referred to as  $\pi_k^{od}$ . All paths connecting  $o$  and  $d$  form a set  $K^{od}$ .

For defining optimality with respect to stochastic routing we follow [43]. The subsequent definitions and propositions are based thereon. The optimal path with respect to a time budget  $b$  is defined as follows.

**Definition 4.1.** A path  $k^{od}$  is  $b$ -reliable in  $K^{od}$  if and only if  $u_k^{od}(b) \geq u_l^{od}(b)$  for all  $l$ , where  $u_k^{od}(b) = P(\pi_k^{od} \leq b)$  denotes the CDF of  $\pi_k^{od}$ .

As the time budget  $b$  depends on the length of the path, it is advantageous to define optimality with respect to a given on-time reliability  $\alpha$  instead.

**Definition 4.2.** A path  $k^{od}$  is  $\alpha$ -shortest in  $K^{od}$  if and only if  $v_k^{od}(\alpha) \leq v_l^{od}(\alpha)$  for all  $l$ , where  $v_k^{od} = (u_k^{od})^{-1}$  is the inverse CDF.

Definition 4.2 enables travelers to budget travel time in order to obtain a desirable reliability of on-time arrival and, thus, better describes their risk

averse routing behavior. Both definitions for the optimal path are, however, equivalent.

**Proposition 4.1.** *A path  $k^{od}$  is  $\alpha$ -shortest if and only if it is  $b$ -reliable for  $b = v_k^{od}(\alpha)$ .*

*Proof.*  $\alpha$ -shortest implies that  $b_k \equiv v_k^{od}(\alpha) \leq b_l \equiv v_l^{od}(\alpha)$  for all  $l$ . This implies that  $u_k^{od}(b_k) \geq u_l^{od}(b_k) \geq u_l^{od}(b_l)$  for all  $l$ . The last inequality holds as  $u_l^{od}$  is monotone and the first inequality implies  $b$ -reliability.  $\square$

## 4.3 Methodology

We have introduced the DDC Model in the previous chapter. It enables us to estimate travel time distribution for a path by using historical travel times on link level. For applying the DDC Model one does not need to make assumptions on the travel time distributions, instead one can work directly with historical travel times. In addition, the DDC Model incorporates correlation of travel times between successive links. These are precisely the two shortcomings of the state-of-the-art SSP problem we want to tackle for developing a stochastic router. We propose a stochastic router that gives an a priori solution. For that reason we calculate alternate routes for the specified origin – destination pair. Next, we will generalize the DDC Model and apply it to each alternate route in order to determine the  $\alpha$ -shortest path.

### 4.3.1 Calculating Alternate Routes

The requirements for the alternate routes are that they need to be geometrically different but also feasible regarding travel time. Using Yen's algorithm we can obtain a set of  $k$  shortest paths. Here, we work with deterministic travel times as in the traditional SPP in order to obtain the paths, i.e. we take the expectation value of the travel time of each link as its cost. However, the paths obtained by Yen's algorithm are very similar due to the procedure of the algorithm itself. In order to obtain a set of alternate routes that fulfil our requirements, we propose the following approach: We calculate  $k$  shortest paths using Yen's algorithm for a large  $k$ , which depends on the distance between origin and destination. Then, we filter these paths by introducing a *similarity score*  $\gamma \in [0, 1]$  which gives the portion of same links between two paths. The procedure for obtaining alternate routes is described in Algorithm 3 with time complexity  $O(N)$ .

**Algorithm 3:** AlternateRoutes ( $o, d, k, \gamma$ )

---

```

 $K^{od} \leftarrow \text{Yen}(o, d, k)$  // Compute  $k$  shortest paths between  $o$  and  $d$  using
Yen's algorithm ;
 $A[0] \leftarrow k_1^{od}$  //  $A$  will be the set of alternate routes ;
 $S \leftarrow L_1$  // Save all  $n$  links  $L_1 = \{l_{11}, \dots, l_{1n}\}$ , of the shortest path  $k_1^{od}$ 
where  $S$  will be the set of used links ;
for  $i \in \text{range}(2, k)$  do
    Compare all  $m$  links  $L_i = \{l_{i1}, \dots, l_{im}\}$  of path  $k_i^{od}$  with the links
    stored in  $S$  and calculate portion of same links  $x(L_i, S)$  ;
    if  $x(L_i, S) \leq \gamma$  then
        Append  $k_i^{od}$  to  $A$  ;
        Append  $L_i$  to  $S$  ;
    end
end
return  $A$ 

```

---

As the parameters  $k$  and  $\gamma$  are highly dependent on the specific origin – destination pair, it is not possible to derive a generalized expression for them. They need to be estimated for each request. For research purposes this approach works just fine. However, for real world application it is neither realistic to expect from the user to define these parameters nor to conduct an estimation procedure for each request.

There are already providers such as Camvit ([1]) and HERE ([2]) which offer a service for computing alternate routes. As this is their business model, the underlying algorithms are not available publicly. Because these alternate routes already meet our requirements without filtering we will use the API provided by HERE for that purpose, such that

$$A \leftarrow \text{HEREAPI}(o, d, k).$$

We can either specify a  $k$  to obtain  $k$  alternate routes or request all alternate routes that are possible.

### 4.3.2 Generalized DDC Model

We have introduced the DDC Model for estimating path travel time distribution for ten links in the previous chapter. Now, we need to generalize the DDC Model in order to work with an arbitrary number of links, which is illustrated in Figure 4.1. For that purpose, we need to generalize the values for Kendall's tau for each iteration, as we want to avoid performing a correlation analysis in

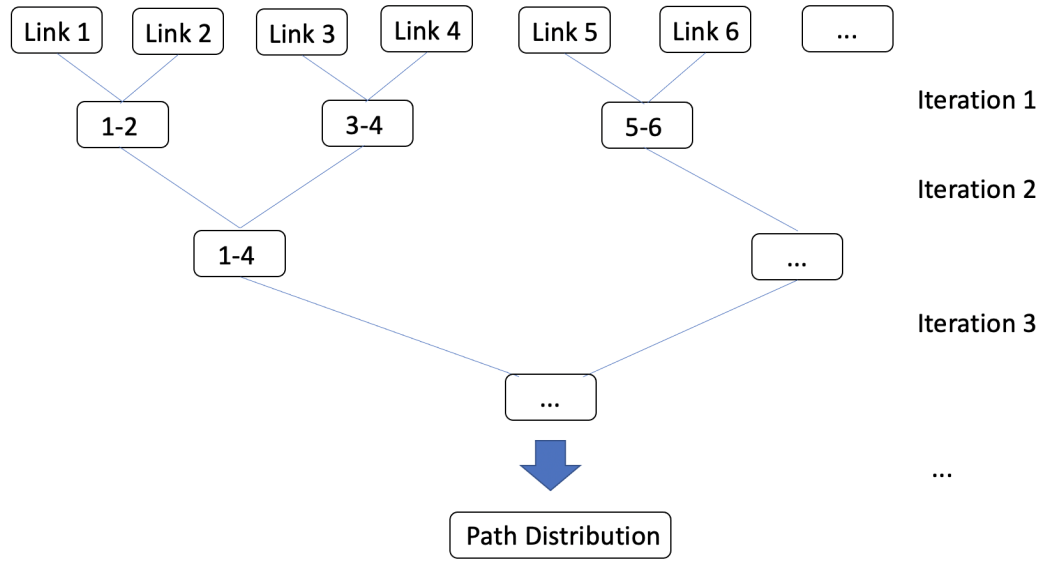


FIGURE 4.1: DDC Model for an arbitrary number of links.

real world application. Furthermore, the data we have for a though movement of an arterial is limited due to privacy reasons as we have mentioned earlier. It is not possible to analyze correlation for iterations higher than Iteration 3 as we only have travel time data for a path comprised of ten links. Thus, we propose an extrapolation method for estimating the correlation for higher iterations.

The average value for Kendall's tau between two successive links is given by

$$\bar{\tau}_{\text{pair}} = \frac{1}{k} \sum_{i=1, j=i+1}^k \tau(\mathbf{x}_i, \mathbf{x}_j)$$

where  $k = n - 1$ ,  $n$  is the number of links, and  $\mathbf{x}_i = \{x_{i1}, \dots, x_{il}\}^T$  are the  $l$  empirical travel times for link  $i$ . In Figure 4.2a we show the correlation matrix for the ten link pairs and we have  $\bar{\tau}_{\text{pair},1} = 0.59$  for Iteration 1. Next, we add the empirical travel times for the individual links order to obtain travel times for "1-2", ..., "9-10". The correlation between "1-2", ..., "9-10" are shown in Figure 4.2b. The corresponding  $\bar{\tau}_{\text{pair},2} = 0.36$  can then be used for Iteration 2. We will use the same value for the intermediate step within Iteration 2 for the sake of simplicity. Analogously we obtain  $\bar{\tau}_{\text{pair},3} = 0.24$  for Iteration 3 as shown in Figure 4.2c. We can then extrapolate a function based on our data points. An exponential function showed the best fit, which is shown in Figure 4.3. This correlation function provides a value of Kendall's tau for each Iteration within the DDC Model, which enables us to use the DDC Model for

an arbitrary number of links.

As our goal is to develop a stochastic router for real world application, we aim for a best possible efficiency of the model. From Figure 4.3 we can observe that the correlation decreases with the number of iterations and becomes marginal after Iteration 5 as  $\tau(\text{Iteration } 5) < 0.15$ . For that reason we propose an extension of the DDC Model, where we use traditional convolution starting from Iteration 5 as the correlation at that point becomes neglectable. We can calculate discrete convolution efficiently using Fast Fourier Transformation (FFT) and denote this extension of the DDC Model as *DDC-FFT Model*, which is illustrated in Figure 4.4.

Pseudo-code for the generalized DDC Model can be found in Algorithm 4 with the following notation:

- $S = \{X_1, \dots, X_n\}$ : Set of travel times  $X_i$  for link  $i$  for path consisting of  $n$  links.
- $(*)^\tau$ : Operator of the DDC with a specified  $\tau$ .
- $|S|$ : Number of elements in  $S$ .
- $a \bmod b = 0$ : Denotes that  $a$  is an integer multiple of  $b$ .

### 4.3.3 Stochastic Router Formulation

Now we can formulate our stochastic router. Given  $G(N, L, P)$ ,  $s, d \in G$ ,  $k \in \mathbb{N}^+$  and  $\alpha \in [0, 1]$  we compute the  $\alpha$ -shortest path from  $s$  to  $d$ . First, we calculate the alternate routes. Then, we apply the DDC Model on each route. Finally, we select the optimal path. The procedure of the stochastic router is summarized in Algorithm 6.

## 4.4 Case Study

We have already evaluated the DDC Model in the previous chapter. Hence, the purpose of this case study is the illustration of applying the DDC Model in a real world scenario. We have chosen an origin - destination pair in the inner city of Munich, Germany, which is shown in Figure 4.5. Alternate routes have been calculated and for each route the DDC Model was applied. Yen's algorithm was applied for  $k = 20$  paths and a similarity score of  $\gamma = 0.5$  was chosen as it showed a good fit for obtaining geometrically different alternate

**Algorithm 4:** DDCModel ( $S$ )

---

```

 $c \leftarrow 0$  // Iteration  $c$  ;
while  $|S| > 1$  do
     $c \leftarrow c + 1$  ;
     $\tau \leftarrow f_\tau(c)$  ;
     $S_{\text{new}} \leftarrow \emptyset$  ;
    if  $|S| \bmod 2 = 0$  then
        for  $(A, B) \in S$  do
             $Z \leftarrow A \stackrel{\tau}{*} B$  ;
            Append  $Z$  to  $S_{\text{new}}$  ;
        end
    else
        for  $(A, B) \in S \setminus \{X_{|S|-2}, X_{|S|-1}, X_{|S|}\}$  do
             $Z \leftarrow A \stackrel{\tau}{*} B$  ;
            Append  $Z$  to  $S_{\text{new}}$  ;
        end
         $Z \leftarrow X_{|S|-2} \stackrel{\tau}{*} X_{|S|-1}$  ;
         $Z \leftarrow Z \stackrel{\tau}{*} X_{|S|}$  ;
        Append  $Z$  to  $S_{\text{new}}$  ;
    end
     $S \leftarrow S_{\text{new}}$ 
end
 $X_{\text{path}} \leftarrow S[1]$  //  $X_{\text{path}}$  is the path travel time ;
return  $X_{\text{path}}$ 

```

---

**Algorithm 5:** DDC-FFTModel ( $S$ )

---

```

 $c \leftarrow 0$  ;
for  $c \in [1, 4]$  do
     $S \leftarrow \text{DDCModel}(S)$  ;
end
 $X_{\text{path}} = \mathcal{F}^{-1}(\prod_{i=1}^m \mathcal{F}\{S[i]\})$  ;
return  $X_{\text{path}}$ 

```

---

**Algorithm 6:** StochasticRouter ( $G, s, d, k, \alpha$ )

---

```

 $B \leftarrow \emptyset$  ;
 $A \leftarrow \text{HEREAPI}(s, d, k)$  ;
for  $A_i \in \{A_0, \dots, A_n\}$  do
     $X_{A_i} \leftarrow \text{DDC-FFTModel}(S_{A_i})$  ;
     $b_i \leftarrow v_i(\alpha)$  ;
    Append  $b_i$  to  $B$  ;
end
 $b_{\text{optimal}} = \min\{B\}$  ;
return Optimal path

```

---



routes. Figure 4.6 shows the PDFs for each route and in Table 4.1 the travel times for different values of  $\alpha$  are listed.

TABLE 4.1: Results for Case Study for different values of  $\alpha$ .

$\alpha$	Path 1	Path 2	Path 3
0.5	15 min	18min	20min
0.7	16 min	20 min	22 min
0.9	18 min	22 min	25 min
1	32 min	38 min	41 min

In order to demonstrate the use case for finding the  $b$ -reliable path, which is the path that maximizes the probability of arriving on time, we set our deadline to  $b = 20$  min. Therefore, Path 1 is the path that maximizes the probability of arriving within  $b$ .

TABLE 4.2: Probability of arriving within  $b = 15$ min for each path of the case study.

	Path 1	Path 2	Path 3
Probability of arriving on time	0.96	0.65	0.40

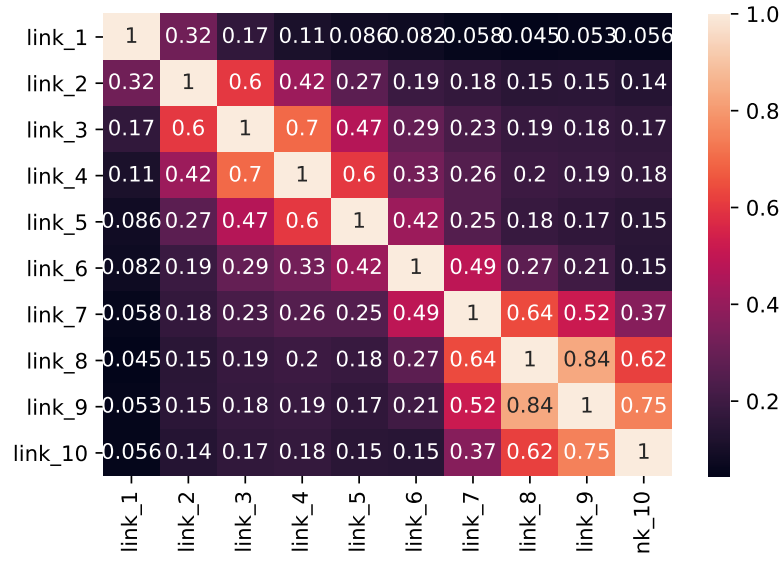
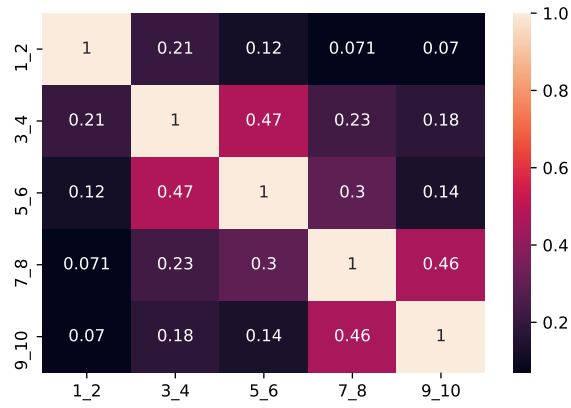
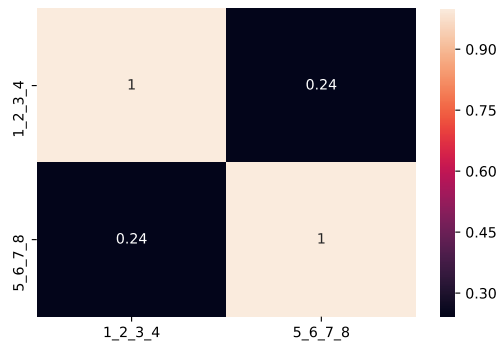
(A)  $\bar{\tau}_{\text{pair},1} = 0.59$ (B)  $\bar{\tau}_{\text{pair},2} = 0.36$ (C)  $\bar{\tau}_{\text{pair},3} = 0.24$ 

FIGURE 4.2: Correlation matrices.

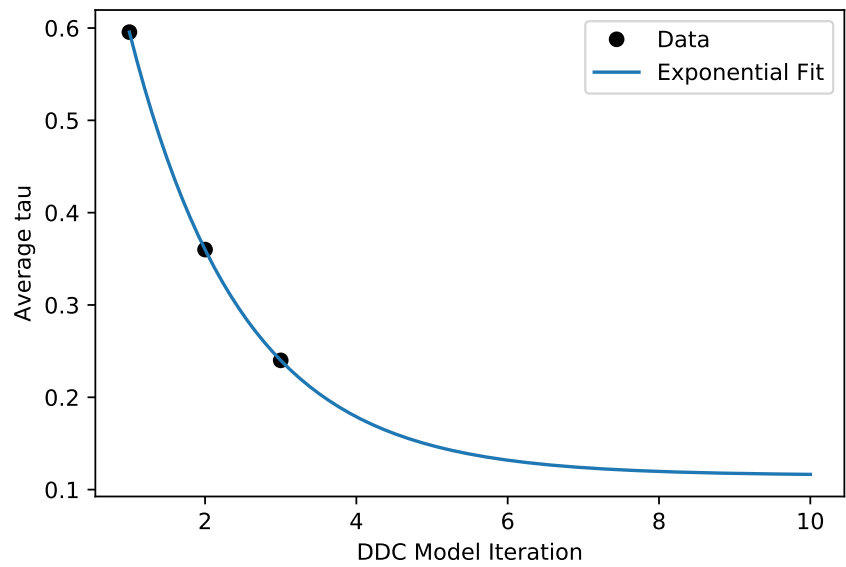


FIGURE 4.3: Correlation function for the DDC Model.

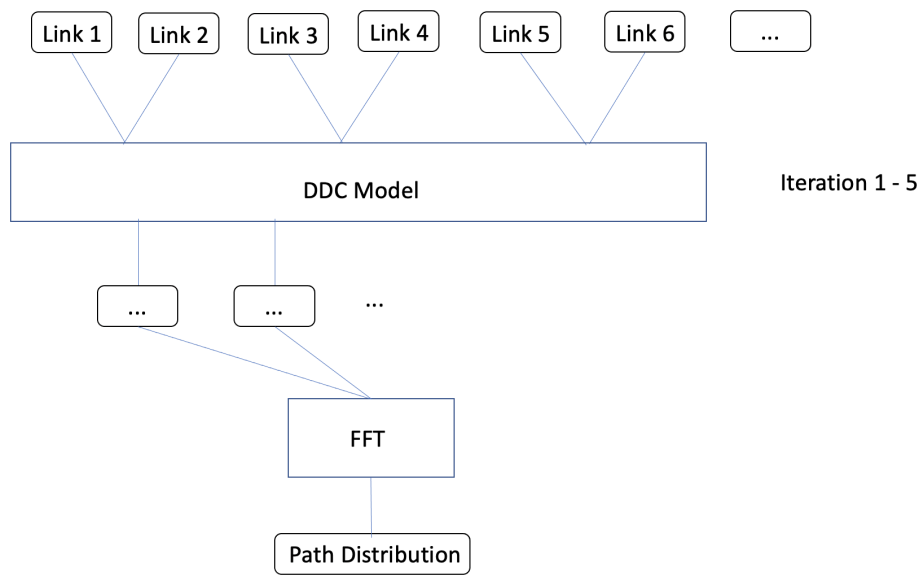


FIGURE 4.4: Schematic illustration of the DDC-FFT Model.

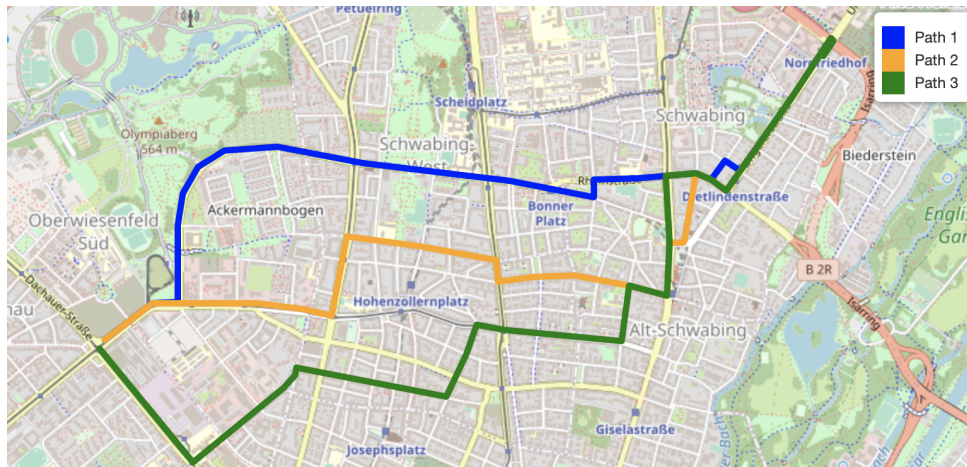


FIGURE 4.5: Case Study for DDC-Routing in the inner city of Munich, Germany.

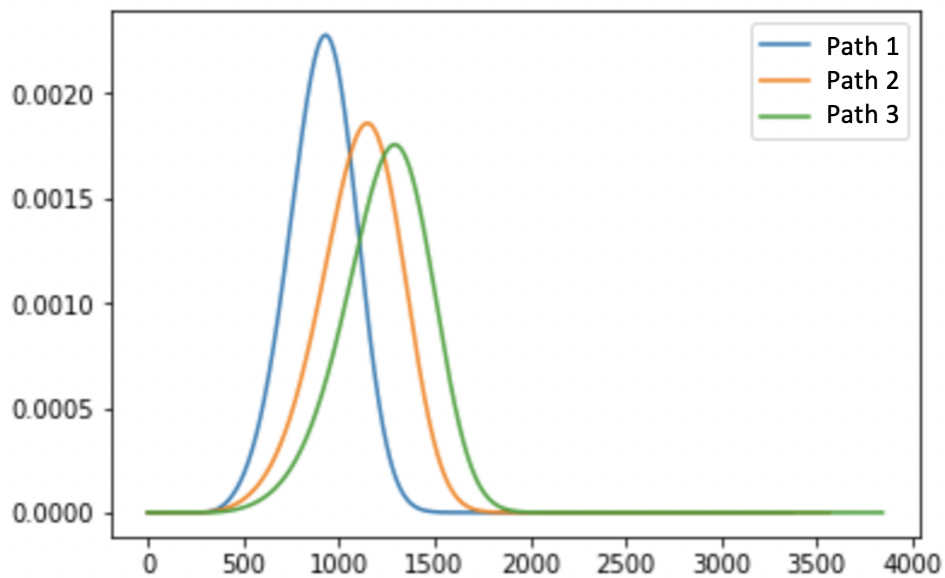


FIGURE 4.6: Case Study for DDC-Routing in the inner city of Munich, Germany.

## Chapter 5

# Routing Optimized for Autonomous Driving

So far, we have developed a methodology for assessing travel time reliability and used this as a basis to design a stochastic router. We will now extend the concept of routing to optimize it for AD. The motivation for this procedure has already been elucidated in the Introduction. Related work on routing has been discussed in the previous chapter. To the best of our knowledge, there is no approach in state-of-the-art literature that discusses a the problem of finding an optimal route with respect to AD.

This thesis provides two solution approaches for the just mentioned problem. The initial situation is the same as for the stochastic router. We calculate a set of alternate routes for one origin – destination pair. From the set of alternate routes our goal is to find the optimal route with respect to AD. The first approach is purely subjective, i.e. based on our personal preferences. In the second one, we use tools provided by the area of multiobjective optimization. There, we aim to avoid subjectivity for the most part. We introduce the methodology for both solution approaches. Then, we conduct a case study applying both models for real world scenarios. The results are compared and interpreted. In the end of this chapter, we propose an integrated approach of a stochastic router optimized for AD. As both the model of stochastic routing and the model of optimized routing for AD will be evaluated separately by then, an additional evaluation for their combination is redundant.

## 5.1 Methodology

### 5.1.1 A Subjective Approach

In order to find one optimal route with respect to AD from a set of alternate routes, we need to clarify the relevant criteria for the selection process as a prior. The first criteria is straightforward, it is the estimated travel time. The second criteria is the fraction of estimated travel time that cannot be driven autonomously compared to the estimated total travel time. The third and last criteria is how often the driver is requested by the vehicle to take over control when in autonomous mode. This happens before reaching a road section where road clearance is deactivated. Based on these three criteria we need to rate the alternate routes to find the best one. For that purpose, our idea is to introduce a *quality* for each path, which enables such a rating. In the following, we describe how we can obtain a value for this quality.

Let us first clarify the notation:

- $\mathbf{x} = \{1, 2, \dots, l\}^T$ : Vector containing all  $l \in \mathbb{R}_n^+$  alternate paths, which are represented as an integer ordered by the outcome of the alternate routing application.
- $t_{\text{total}}(\mathbf{x})$ : Estimated total travel time for each alternate path.
- $t_{\text{manual}}(\mathbf{x})$ : Estimated travel time that needs to be driven manually for each alternate path, or in other words, estimated travel time that cannot be driven autonomously.
- $n_{\text{tor}}(\mathbf{x})$ : Number of take over requests for each alternate path., i.e. requests from the vehicle to the driver to take over control of the vehicle in autonomous mode.

For each criteria  $t_{\text{total}}$ ,  $t_{\text{manual}}$ , and  $n_{\text{tor}}$ , we introduce a *subquality*  $q_1$ ,  $q_2$ , and  $q_3$ , respectively. Each subquality is represented as a function that is designed to map the value of the respective criteria to a value of satisfaction of the user. The functions were obtained in an iterative process by describing our own preferences.

#### Derivation of $q_1$

When selecting an alternate route, how much delay compared to the shortest route would we accept and still be satisfied? We denote the delay as  $d$ , and

define it as  $d(x_i) := t_{\text{total}}(x_i)/t_{\text{shortest}}$ , where  $x_i, i \in \{1, 2, \dots, n\}$  is the route currently under consideration and  $t_{\text{shortest}} = \min\{t_{\text{total}}(\mathbf{x})\}$  i.e. travel time of the shortest route. Then, for the shortest route it holds  $d = 0$  and we set  $q_1(d = 0) = 1$ , i.e. the value that represents maximum satisfaction. We assume a linear decrease in satisfaction with increasing delay and set  $q_1(d = 0.25 \cdot t_{\text{shortest}}) = 0$  and  $q_1(d > 0.25 \cdot t_{\text{shortest}}) < 0$ , leading to

$$q_1(d) = 1 - 4 \cdot d,$$

i.e. with a delay up to 25 % compared to the shortest route we are still satisfied, afterwards we penalize.

### Derivation of $q_2$

The more travel time can be driven autonomously, the more satisfied we are. Therefore, we simply define

$$q_2(t_{\text{total}}, t_{\text{manual}}) = \frac{t_{\text{total}} - t_{\text{manual}}}{t_{\text{total}}}.$$

### Derivation of $q_3$

Analogously to the derivation of  $q_1$ , we assume a linear decrease in satisfaction with an increasing number of take over requests. We set  $q_3(n_{\text{tor}}/\text{hr} = 0) = 1$  for maximum satisfaction, where  $n_{\text{tor}}/\text{hr}$  denotes the the number of take over requests per hour. We set  $q_3(n_{\text{tor}}/\text{hr} = 5) = 0$  and obtain

$$q_3(n_{\text{tor}}/\text{hr}) = 1 - 0.2 \cdot n_{\text{tor}}/\text{hr}.$$

Now that we have derived each subquality we define the quality of each alternate path  $x_i$ , denoted as  $Q$ , as a weighted sum

$$Q(x_i) = \omega_1 q_1(x_i) + \omega_2 q_2(x_i) + \omega_3 q_3(x_i),$$

where  $\omega = \{\omega_1, \omega_2, \omega_3\}$ , commonly chosen so that  $\sum_{j=1}^3 \omega_j = 1$ , are the respective weights. We set  $\omega = \{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\}$  as we want to include each criteria with equal impact.

Clearly, this approach is highly subjective. We have made several assumptions

and have defined arbitrary parameters according to our own preferences. In order to further evaluate and develop this model, we would need to conduct surveys with a larger sample size, which is not within the scope of this thesis. Nevertheless, we included this approach in order to compare it with the formulation as a multiobjective optimization problem, where we tried to avoid subjectivity at all.

### 5.1.2 Multiobjective Optimization Problem Formulation

For the second solution approach, we also start by calculating alternate routes for one origin – destination pair. Our criteria for finding the optimal route from the set of alternate routes stay the same, i.e.  $t_{\text{total}}$ ,  $t_{\text{manual}}$ , and  $n_{\text{tor}}$ . Optimization problems that involve multiple conflicting objectives are referred to as multiobjective optimization problems. As this is the case for our problem, we will now derive a formulation in the sense of multiobjective optimization based on [10], which takes the form

$$\begin{aligned} & \text{minimize} \quad \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})\} \\ & \text{subject to} \quad \mathbf{x} \in S, \end{aligned} \tag{5.1}$$

with objective functions  $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$  and decision vectors  $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ , which belong to the nonempty feasible region  $S \subset \mathbb{R}^n$ . Objective vectors are images of decision vectors and are comprised of objective values  $\mathbf{z} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x}))^T$ . The image of the feasible region in the objective space is referred to as objective region  $Z = \mathbf{f}(S)$ .

Objective vectors are considered as optimal if none of their components can be improved without diminution of at least one of the other components. A decision vector is referred to as *Pareto optimal* if there is no other  $\mathbf{x} \in S$  such that  $f_i(\mathbf{x}) \leq f_i(\mathbf{x}')$  for all  $i = 1, \dots, k$  and  $f_j(\mathbf{x}) < f_j(\mathbf{x}')$  for at least one index  $j$ . The set of Pareto optimal decision vectors is denoted as  $P(S)$ . Thus, an objective vector is Pareto optimal if its decision vector is Pareto optimal. The set of Pareto optimal objective vectors is denoted as  $P(Z)$ . The set of Pareto optimal solutions is a subset of the set of weakly Pareto optimal solutions. A decision vector  $(\mathbf{x}')$  is weakly Pareto optimal if there is no other  $\mathbf{x} \in S$  such that  $f_i(\mathbf{x}) < f_i(\mathbf{x}')$  for all  $i = 1, \dots, k$ . Analogously two sets are denoted corresponding to decision and objective spaces by  $WP(S)$  and  $WP(Z)$ , respectively.



In order to obtain one optimal solution from the set of Pareto optimal solutions, there are several methods. The key difference considering these methods is, if there is a Decision Maker (DM) bringing in subjective preferences into the solution process, or not. These methods are divided into two classes. There are *interactive methods*, where an iterative solution algorithm is generated and applied repeatedly. After each iteration, information is given to the DM, who can then specify his preferences. It is the most expensive class of methods and does not fit our purpose to find an optimal route for autonomous vehicles. We need more straightforward approaches, which are provided by the class of *non-interactive methods*. Depending on the role of the DM the non-interactive methods are divided into three categories. If there is no DM involved in the solution process, we refer to this case as *no-preference method*. Thus, the goal is to find a compromise solution without any preferences. One example for a no-preference method is the method of global criterion [69]. The idea is to minimize the distance between the feasible region and an ideal objective vector  $\mathbf{z}^{\text{ideal}}$ . The problem is formulated and solved as

$$\begin{aligned} & \text{minimize} \quad ||f_i(\mathbf{x}) - z_i^{\text{ideal}}|| \\ & \text{subject to} \quad \mathbf{x} \in S, \end{aligned}$$

where  $||\cdot||$  can be any  $L_p$  norm. In [41] it is shown that the choice of the norm affects the obtained solution. Furthermore, the objective functions need to be scaled to a uniform dimensionless space. Finding such a standardization for our problem is not straightforward. In [58] it is mentioned, that the selected standardization affects the ranking. We would rather avoid such a procedure and apply a solution method without needing to manipulate our objective functions.

In the remaining cases, the DM is included in the decision process. In *a priori methods*, the DM sets preferences beforehand and the goal is to find one Pareto optimal solution that satisfies them as well as possible. Here, the difficulty is that the DM may not be aware of the limitations of the problem and, therefore, his expectations may be too optimistic or pessimistic. One example is the *value function method*. If the DM knows an explicit mathematical formulation for the value function and if that function is able to incorporate all of the DM's preferences, the problem is solved by

$$\begin{aligned} & \text{maximize} \quad v(\mathbf{f}(\mathbf{x})) \\ & \text{subject to} \quad \mathbf{x} \in S. \end{aligned}$$

However, this is not the case for our problem.

There is one category of non-interactive methods left, which is the class *a posteriori methods*. Here, the set of Pareto optimal solution is obtained at first, and the DM then selects the most preferred one. The DM obtains an overview of varying solutions, and articulates his preference information afterwards. *A posteriori methods* show the best fit to our problem. Now, we have to find one method in this class, where we do not have to manipulate our objective functions. The Weighted Product Model (WPM), see [58, 59], provides just this. Before we give its definition, we first formulate our problem.

We define our optimization model as follows. The decision vector  $\mathbf{x} = \{1, 2, \dots, l\}^T$  remains the same as in the subjective solution approach. The objective functions are formulated as

$$\begin{aligned} f_1(\mathbf{x}) &= t_{\text{total}}(\mathbf{x}) \\ f_2(\mathbf{x}) &= t_{\text{manual}}(\mathbf{x}) \\ f_3(\mathbf{x}) &= n_{\text{tor}}(\mathbf{x}). \end{aligned}$$

Following Eq. (5.1) our optimization problem takes the form

$$\begin{aligned} &\text{minimize} \quad \{f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x})\} \\ &\text{subject to} \quad \mathbf{x} \in S. \end{aligned}$$

First, we obtain the set of Pareto optimal routes from the set of alternate routes. This process is straightforward using the definition of Pareto optimality. Then, we need to select one optimal route from the set of Pareto optimal routes. This is where the WPM comes into play. It enables us to compare two Pareto optimal routes with each other and is defined as

$$R(x_k/x_l) = \prod_{j=1}^3 (f_j(x_k)/f_j(x_l))^{\omega_j},$$

where  $\omega_i$  is the weight for the  $i$ -th criteria with  $\sum_{j=1}^3 \omega_j = 1$ . We set  $\boldsymbol{\omega} = \{\omega_1, \omega_2, \omega_3\} = \{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\}$  for the same reason as in the previous section. If  $R(x_k/x_l) < 1$ , then  $x_l$  is better than  $x_k$ . Conversely, if  $R(x_k/x_l) > 1$ , then  $x_k$  is better than  $x_l$ . For a manageable amount of Pareto optimal alternatives (which is the case for our alternative routing as we will see when conducting case studies) we can simply compare all of the alternatives and obtain the optimal

one. Note, that the WPM does not require any form of standardization. We can simply plug in the values of our objective functions, without needing to worry about units.

## 5.2 Case Studies

For evaluation, we apply both proposed methodologies to three origin - destination pairs in Germany, namely Stuttgart - Cologne, Munich - Frankfurt, and Frankfurt - Dresden. For obtaining alternate routes we first used Algorithm 3, that we developed and applied in the previous chapter. However, this approach was not feasible as the distance between two cities is very large compared to the distance between origin and destination in an urban scenario. Even by setting  $k = 100$  there was no obvious second alternate route available yet. Therefore, we used an already existing alternative with HERE ([2]) which provides alternate routes and their corresponding travel time information.

For the sake of simplicity, we assume activated road clearance for all freeway facilities, also referred to as road sections of *functional class 1*. In order to represent the dynamics of the features affecting road clearance, we include incidents restricting road clearance, motivated by actual traffic information data. Those incidents can be any of which cause a withdrawal of road clearance activation. Thus, they are not specified, and referred to as *restrictions* in the following. For each case study, we compare the results of three scenarios with different traffic situations.

### 5.2.1 Munich – Frankfurt

Our first case study is to find an optimal route for AD from Munich to Frankfurt. Figure 5.1 shows the alternate routes available for this origin – destination pair, and the relevant information for our criteria can be found in Table 5.1. The results obtained by the quality functions are shown in Table 5.2 and the results obtained by the multiobjective optimization are illustrated in Table 5.3. We can observe that for every scenario both approaches yield the same results. In Scenario I Path 1 is optimal, in Scenario II Path 2 is optimal, and in Scenario III the optimal path is Path 3.

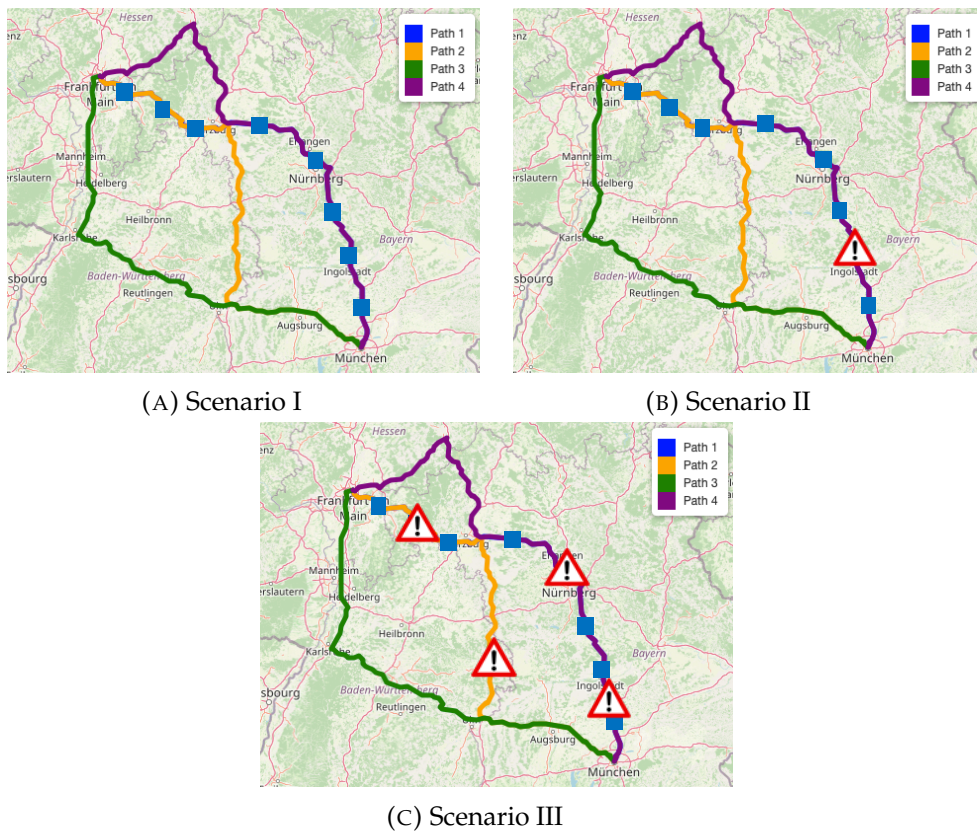


FIGURE 5.1: Munich – Frankfurt: Three scenarios with different restrictions for activation of road clearance. The warning triangle symbolizes the restriction on the respective path.

	Path	Total Time	Manual Driving Time	TORs
<b>Scenario I</b>	1	3h 38min	0h 26 min	1
	2	3h 34min	0h 39min	2
	3	4h 17min	0h 34min	2
	4	4h 17min	0h 38min	1
<b>Scenario II</b>	1	3h 38min	0h 56 min	2
	2	3h 34min	0h 39min	2
	3	4h 17min	0h 34min	2
	4	4h 17min	1h 08min	2
<b>Scenario III</b>	1	3h 38min	1h 44min	5
	2	3h 34min	1h 27min	4
	3	4h 17min	0h 34min	2
	4	4h 17min	1h 35 min	3

TABLE 5.1: Munich – Frankfurt: Relevant information on each path shown in Figure 5.1.

<b>Scenario I</b>		<b>Scenario II</b>		<b>Scenario III</b>	
Path	Quality	Path	Quality	Path	Quality
1	0.85	1	0.72	1	0.40
2	0.77	2	0.77	2	0.53
3	0.56	3	0.56	3	0.56
4	0.64	4	0.52	4	0.40

TABLE 5.2: Munich – Frankfurt: Results quality functions.

### 5.2.2 Stuttgart – Cologne

The second case study is to find an optimal route for AD from Stuttgart to Cologne. Figure 5.2 shows the alternate routes and the relevant information for our criteria can be found in Table 5.4. The results obtained by the quality functions are shown in Table 5.5 and the results obtained by the multiobjective optimization are illustrated in Table 5.6. For every scenario both approaches yield the same results. In Scenario I Path 4 is optimal, in Scenario II Path 1 is optimal, and in Scenario III the optimal path is Path 3.

### 5.2.3 Frankfurt – Dresden

Here, we take Stuttgart and Cologne as our origin – destination pair. Figure 5.3 shows the alternate routes and the relevant information for our criteria can

Scenario I		Scenario II		Scenario III	
Pareto	WPM	Pareto	WPM	Pareto	WPM
1, 2	$R_{1,2} = 0.70$	2, 3	$R_{2,3} = 0.98$	2, 3, 4	$R_{2,3} = 1.60$
					$R_{2,4} = 1.01$
					$R_{3,4} = 0.63$

TABLE 5.3: Munich – Frankfurt: Results multiobjective optimization WPM. We denoted  $R_{j,k} := R(x_j/x_k)$ .

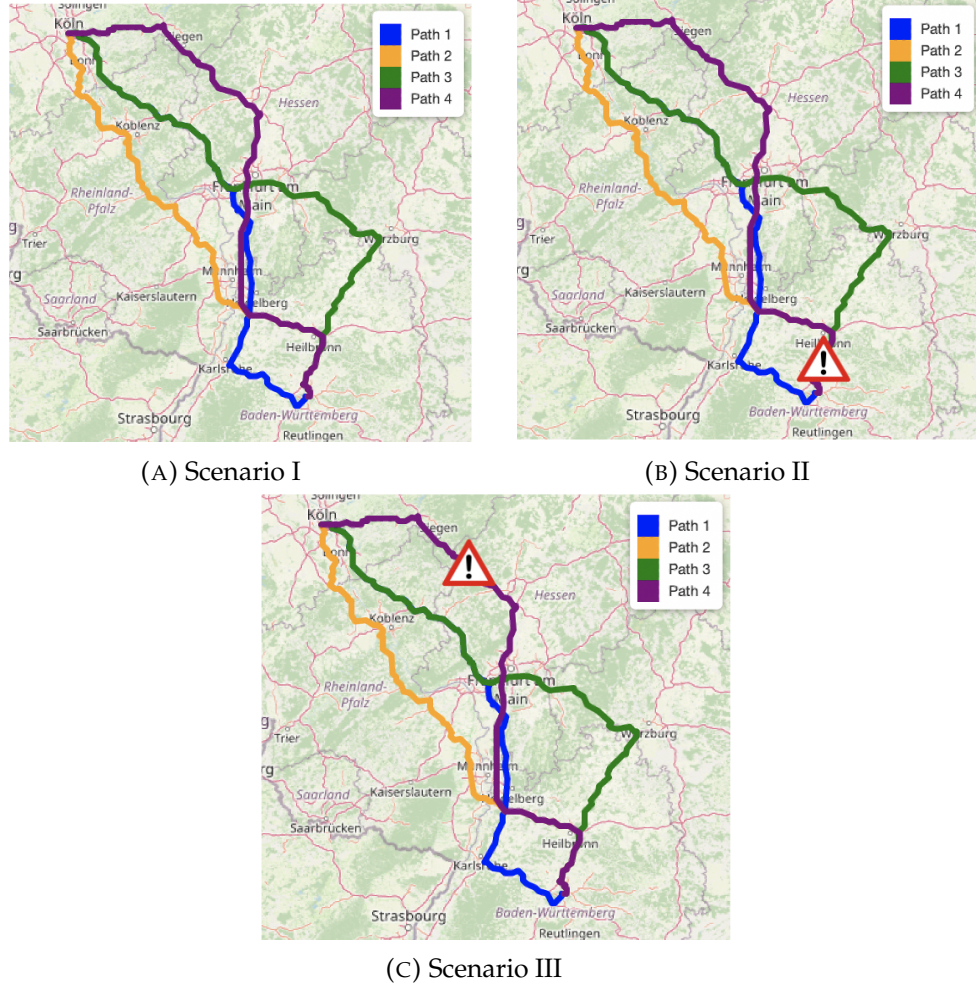


FIGURE 5.2: Munich – Frankfurt: Three scenarios with different restrictions for activation of road clearance. The warning triangle symbolizes the restriction on the respective path.

	Path	Total Time	Manual Driving Time	TORs
<b>Scenario I</b>	1	4h 19min	0h 36min	1
	2	4h 42min	0h 41min	1
	3	4h 08min	0h 38min	1
	4	4h 00min	0h 38min	1
<b>Scenario II</b>	1	4h 19min	0h 36min	1
	2	4h 42min	1h 18min	2
	3	4h 08min	0h 54min	2
	4	4h 00min	1h 15min	2
<b>Scenario III</b>	1	4h 19min	0h 36min	1
	2	4h 42min	0h 41min	1
	3	4h 08min	0h 38min	1
	4	4h 00min	0h 56min	2

TABLE 5.4: Stuttgart – Cologne: Relevant information on each path shown in Figure 5.1.

<b>Scenario I</b>		<b>Scenario II</b>		<b>Scenario III</b>	
Path	Quality	Path	Quality	Path	Quality
1	0.77	1	0.77	1	0.77
2	0.67	2	0.54	2	0.62
3	0.82	3	0.71	3	0.82
4	0.86	4	0.73	4	0.76

TABLE 5.5: Stuttgart – Cologne: Results quality functions.

be found in Table 5.7. The results obtained by the quality functions are shown in Table 5.8 and the results obtained by the multiobjective optimization are listed in Table 5.9. Also in this case study, both approaches yield the same results for every scenario. In Scenario I Path 1 is optimal, in Scenario II Path 2 is optimal, and in Scenario III the optimal path is Path 3.

### 5.2.4 Summary

In all three case studies, the subjective approach with quality functions and the approach using multiobjective optimization yielded the same results. This shows that our definitions for criteria for finding an optimal route for AD are reasonable, as they were the same in both approaches. Furthermore, it can be interpreted as a cross-check that in routing optimality with respect to AD does exist. That is, because we come to the same conclusion with a purely



Scenario I		Scenario II		Scenario III	
Pareto	WPM	Pareto	WPM	Pareto	WPM
1, 3, 4	$R_{1,3} = 1.01$	1, 3, 4	$R_{1,3} = 0.70$	1, 3, 4	$R_{1,3} = 1.01$
	$R_{1,4} = 1.02$		$R_{1,4} = 0.64$		$R_{1,4} = 0.70$
	$R_{3,4} = 1.01$		$R_{3,4} = 0.90$		$R_{3,4} = 0.71$

TABLE 5.6: Stuttgart – Cologne: Results multiobjective optimization WPM. We denoted  $R_{j,k} := R(x_j/x_k)$ .

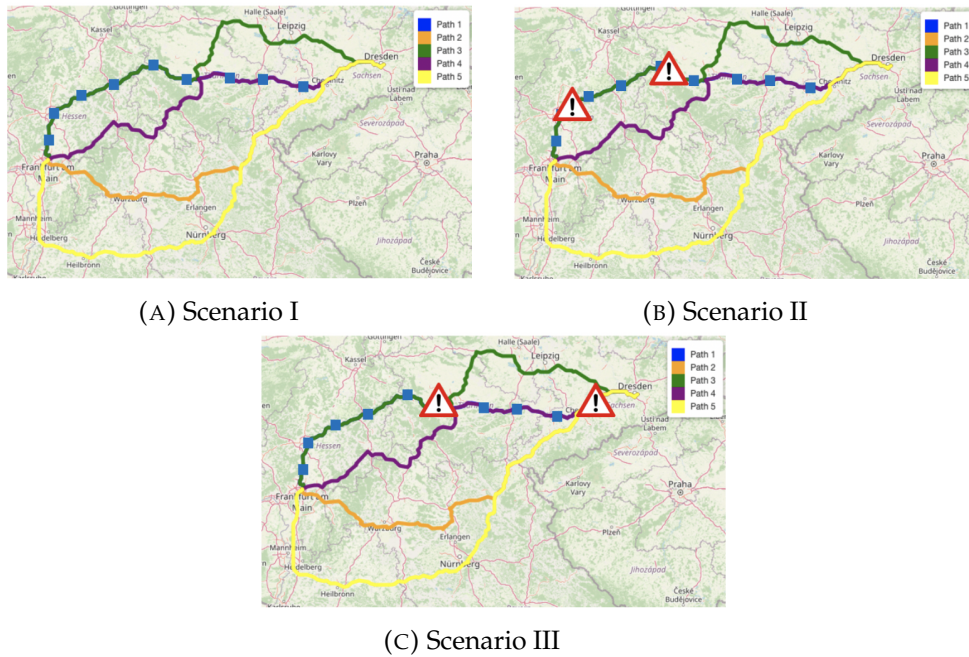


FIGURE 5.3: Frankfurt – Dresden: Three scenarios with different restrictions for activation of road clearance. The warning triangle symbolizes the restriction on the respective path.



	Path	Total Time	Manual Driving Time	TORs
<b>Scenario I</b>	1	4h 26min	0h 23min	1
	2	5h 01min	0h 31min	1
	3	4h 51min	0h 23min	1
	4	5h 20min	1h 47min	2
	5	6h 37min	0h 39min	1
<b>Scenario II</b>	1	4h 26min	0h 59min	3
	2	5h 01min	0h 31min	1
	3	4h 51min	0h 23min	3
	4	5h 20min	1h 47min	2
	5	6h 37min	0h 39min	1
<b>Scenario III</b>	1	4h 26min	1h 20min	2
	2	5h 01min	1h 02min	2
	3	4h 51min	0h 43min	1
	4	5h 20min	2h 44min	3
	5	6h 37min	1h 01min	2

TABLE 5.7: Frankfurt – Dresden: Relevant information on each path shown in Figure 5.1.

subjective approach and an approach, where we avoided subjectivity for the most part. Thus, the sense of optimality based solely on our own preferences does coincide with the notion of optimality provided by the mathematical area of optimization.

The different scenarios reflected how dynamic the activation of road clearance is. As each scenario resulted in a different optimal path, it was shown that the inclusion of AD related optimality into route guidance system for real world application is necessary to ensure an optimal drive with an autonomous

<b>Scenario I</b>		<b>Scenario II</b>		<b>Scenario III</b>	
<b>Path</b>	<b>Quality</b>	<b>Path</b>	<b>Quality</b>	<b>Path</b>	<b>Quality</b>
1	0.89	1	0.68	1	0.73
2	0.73	2	0.73	2	0.61
3	0.78	3	0.57	3	0.76
4	0.50	4	0.50	4	0.36
5	0.45	5	0.45	5	0.34

TABLE 5.8: Frankfurt – Dresden: Results quality functions.

Scenario I		Scenario II		Scenario III	
Pareto	WPM	Pareto	WPM	Pareto	WPM
1, 3	$R_{1,3} = 0.97$	1, 2, 3, 5	$R_{1,2} = 1.70$ $R_{1,3} = 0.97$ $R_{1,5} = 1.57$ $R_{2,3} = 0.57$ $R_{2,5} = 0.92$ $R_{3,5} = 1.62$	1, 3	$R_{1,3} = 1.49$

TABLE 5.9: Frankfurt – Dresden: Results multiobjective optimization WPM. We denoted  $R_{j,k} := R(x_j/x_k)$ .

vehicle.

### 5.3 Combination with Stochastic Routing

So far we have developed a stochastic router and an AD router separately. Now we aim at merging these two approaches in order to obtain a stochastic router optimized for AD, that is introducing travel time reliability in our AD routing framework.

The reliable total travel time can be obtained by applying the DDC Model, as we presented in the previous chapter. For calculating the reliable manual driving time, we apply the DDC Model only on the links, where there is no road clearance activated. The number of take over requests remains the same as in the non-stochastic case, because it is not time-dependent. Figure 5.4

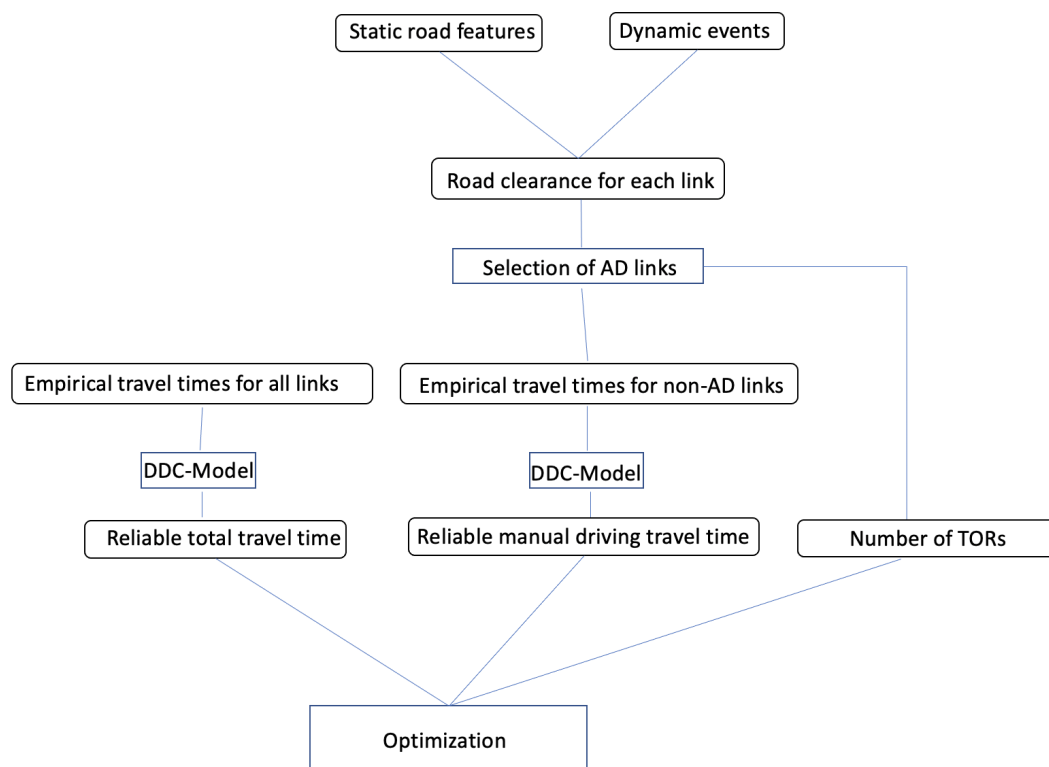


FIGURE 5.4: Schematic illustration of stochastic routing optimized for AD.

illustrates this procedure.

We have already evaluated both the DDC Model and the framework for

AD routing by conducting case studies. Hence, an evaluation of the integrated approach is redundant.

## Chapter 6

# Conclusion and Outlook

### Summary

The aim of this thesis was to develop a stochastic router optimized for autonomous driving. This can be divided into three research objectives.

First a methodology for modelling travel time reliability was presented. Historical travel times were provided by Floating Car Data obtained by a fleet of probe vehicles. A Copula Model was used to aggregate link travel times to a path travel time distribution. Compared to state-of-the-art approaches where link travel times are considered as independent, the Copula Model is able to capture the dependence between links. A case study was conducted for two study sites with different road features, that is one urban arterial and one freeway arterial, in order to evaluate the Copula Model. First link correlation was analyzed and different copulas were compared with respect to their ability to model the respective correlation structure. For both study sites, the Clayton copula performed best and was therefore used in the Copula Model. It was shown that the travel time distribution estimated by the Copula Model is more accurate to the empirical one than the convolution, which assumes independence of link travel times. For the use in a real world stochastic router, the idea of modelling dependence between links using copulas was transferred from the Copula Model and incorporated in the DDC Model. The case study showed that the DDC Model was more accurate than the convolution and more efficient than the Copula Model.

Next, a stochastic router was designed based on the DDC Model. Additional link correlation analysis was performed and an extrapolation method for describing correlation for each Iteration of the generalized DDC model was developed. Alternate routes for one origin – destination pair were calculated and travel time distribution for each route was obtained by a generalized

DDC Model. The  $b$ -reliable or  $\alpha$ -shortest path was then determined. A case study was presented as a use case for real world application.

A routing framework optimized for autonomous driving was developed. Two methodologies were presented. The first one was based on subjective perception and the second one was formulated as an multiobjective optimization problem. Both methods were compared by conducting three case studies and showed analogous results for the optimal path. Eventually, a stochastic router optimized for autonomous driving was designed by integrating the stochastic router into this framework.

## Outlook

Although the available data set was relatively large compared to data used in related literature, there are some limitations to it. As empirical travel times for a through movement of a path consisting of several links is limited due to privacy reasons, the evaluation of both the Copula Model and DDC Model is restricted concerning the length of a path. Future work can focus on obtaining data for longer arterials for further evaluation purposes. For example, a program could be established, where vehicle owners can choose to consent that their FCD are sent out not anonymously but with the possibility to track their total trajectory.

Furthermore, as the stochastic router relies on historical travel time data, a world wide use would require travel time data for each link in the total network. A topic for upcoming research could be to develop novel methods for obtaining data in order to make this possible.

Concerning the routing optimized for AD, a survey could be conducted, where the participants can choose between the alternate routes given the information of each path. This would enable further evaluation for the subjective approach for a larger sample size.

In addition, future work could focus on the implementation of DDC routing algorithm with the goal to further improve efficiency.

# Bibliography

- [1] Camvit: Cambridge vehicle information technology ltd. <http://www.camvit.com>.
- [2] Here: Alternative routes. [https://developer.here.com/documentation/routing/dev\\_guide/topics/public-transport-alternative-routes.html](https://developer.here.com/documentation/routing/dev_guide/topics/public-transport-alternative-routes.html).
- [3] *An Introduction to Copulas*. Springer Series in Statistics, 2005.
- [4] Federal Highway Administration. Travel time reliability: Making it there on time, all the time. [www.ops.fhwa.dot.gov/publications/tt\\_reliability/TTR\\_Report.htm](http://www.ops.fhwa.dot.gov/publications/tt_reliability/TTR_Report.htm).
- [5] Theodore W Anderson. On the distribution of the two-sample cramer-von mises criterion. *The Annals of Mathematical Statistics*, pages 1148–1159, 1962.
- [6] Mohsen Babaei, Mojtaba Rajabi-Bahaabadi, and Afshin Shariat-Mohaymany. Estimation of travel time reliability in large-scale networks. *Transportation Letters*, 8(4):229–240, 2016.
- [7] Chandra R Bhat and Ipek N Sener. A copula-based closed-form binary logit choice model for accommodating spatial correlation across observational units. *Journal of Geographical Systems*, 11(3):243–272, 2009.
- [8] Chandra R Bhat, Ipek N Sener, and Naveen Eluru. A flexible spatially dependent discrete choice model: formulation and application to teenagers' weekday recreational activity participation. *Transportation research part B: methodological*, 44(8-9):903–921, 2010.
- [9] Ronald Newbold Bracewell and Ronald N Bracewell. *The Fourier transform and its applications*, volume 31999. McGraw-Hill New York, 1986.
- [10] Juergen Branke, Kalyan Deb, Kaisa Miettinen, and Slowinski Roman. Multiobjective optimization, interactive and evolutionary approaches [outcome of dagstuhl seminars]. 01 2008.

- [11] Zhiguang Cao, Hongliang Guo, Jie Zhang, and Ulrich Fastenrath. Multiagent-based route guidance for increasing the chance of arrival on time. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [12] Zhiguang Cao, Hongliang Guo, Jie Zhang, Dusit Niyato, and Ulrich Fastenrath. Finding the shortest path in stochastic vehicle routing: A cardinality minimization approach. *IEEE Transactions on Intelligent Transportation Systems*, 17(6):1688–1702, 2015.
- [13] Min Chen, Guizhen Yu, Peng Chen, and Yunpeng Wang. A copula-based approach for estimating the travel time reliability of urban arterial. *Transportation Research Part C: Emerging Technologies*, 82:1 – 23, 2017.
- [14] Peng Chen, Wang Zeng, Min Chen, Guizhen Yu, and Y. Wang. Modeling arterial travel time distribution by accounting for link correlations: a copula-based approach. *Journal of Intelligent Transportation Systems*, 10 2018.
- [15] Donald L Cohn. *Measure theory*. Springer, 2013.
- [16] Jan Dhaene and Marc J Goovaerts. Dependency of risks and stop-loss order 1. *ASTIN Bulletin: The Journal of the IAA*, 26(2):201–212, 1996.
- [17] Elena Di Bernardino and Didier Rulli  re. On an asymmetric extension of multivariate archimedean copulas based on quadratic form. *Dependence Modeling*, 4(1), 2016.
- [18] Edsger W Dijkstra et al. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.
- [19] Lili Du, Srinivas Peeta, and Yong Hoon Kim. An adaptive information fusion model to predict the short-term link travel time distribution in dynamic traffic networks. *Transportation Research Part B: Methodological*, 46(1):235–252, 2012.
- [20] Rick Durrett. *Probability: theory and examples*, volume 49. Cambridge university press, 2019.
- [21] Paul Embrechts, Alexander McNeil, and Daniel Straumann. Correlation and dependence in risk management: properties and pitfalls. *Risk management: value at risk and beyond*, 1:176–223, 2002.



- [22] Mohamed El Esawey and Tarek Sayed. Using buses as probes for neighbor links travel time estimation in an urban network. *Transportation Letters*, 3(4):279–292, 2011.
- [23] Himanshu Bhandoh et al. Lecture notes in computer science, *Stanford University*. May 2017.
- [24] Yueyue Fan, Robert Kalaba, and James Moore. Arriving on time. *Journal of Optimization Theory and Applications*, 127(3):497–513, 2005.
- [25] Yueyue Fan, Robert Kalaba, and James Moore. Shortest paths in stochastic networks with correlated link costs. *Computers & Mathematics with Applications*, 49(9-10):1549–1564, 2005.
- [26] Harary Frank. Shortest paths in probabilistic graphs. *Operations Research*, 17(4):583–599, 1969.
- [27] Nikolaos Geroliminis and Alex Skabardonis. Real time vehicle reidentification and performance measures on signalized arterials. In *2006 IEEE Intelligent Transportation Systems Conference*, pages 188–193, Sep. 2006.
- [28] Rong He, Henry Liu, Alain Kornhauser, and Bin Ran. Temporal and spatial variability of travel time. *UC Irvine: Center for Traffic Simulation Studies*, 2002.
- [29] Kojadinovic I. Maechler M. Hofert, M. and J. Yan. copula: Multivariate dependence with copulas. <https://CRAN.R-project.org/package=copula>, R package version 0.999-19.1, 2018.
- [30] Wassilij Höfdding. Masstabinvariante korrelationstheorie. *Schriften des Mathematischen Instituts und Instituts für Angewandte Mathematik der Universität Berlin*, 5:181–233, 1940.
- [31] Yasunori Iida. Basic concepts and future directions of road network reliability analysis. *Journal of Advanced Transportation*, 33:125 – 134, 03 1999.
- [32] SAE International. Levels of driving automation. <https://www.sae.org/news/2019/01/sae-updates-j3016-automated-driving-graphic>.
- [33] Jeffrey P Kharoufeh and Natarajan Gautam. Deriving link travel-time distributions via stochastic speed processes. *Transportation Science*, 38(1):97–106, 2004.

- [34] Yan J. Kojadinovic, I. Modeling multivariate distributions with continuous margins using the copula r package. *Journal of Statistical Software*, 34(9):1–20, 2010.
- [35] Alexandra Kondyli, Bryan St. George, and Lily Elefteriadou. Comparison of travel time measurement methods along freeway and arterial facilities. *Transportation Letters*, 10(4):215–228, 2018.
- [36] Sejoon Lim, Hari Balakrishnan, David Gifford, Samuel Madden, and Daniela Rus. Stochastic motion planning and applications to traffic. *The International Journal of Robotics Research*, 30(6):699–712, 2011.
- [37] Hans Lint. Online learning solutions for freeway travel time prediction. *Intelligent Transportation Systems, IEEE Transactions on*, 9:38 – 47, 04 2008.
- [38] Henry X Liu and Wenteng Ma. A virtual vehicle probe model for time-dependent travel time estimation on signalized arterials. *Transportation Research Part C: Emerging Technologies*, 17(1):11–26, 2009.
- [39] Pierre Loustau, Catherine Morency, Martin Trépanier, and Louis Gourvil. Travel time reliability on a highway network: estimations using floating car data. *Transportation Letters*, 2(1):27–37, 2010.
- [40] Kate Lyman and Robert L Bertini. Using travel time reliability measures to improve regional transportation planning and operations. *Transportation Research Record*, 2046(1):1–10, 2008.
- [41] Kaisa Miettinen. Nonlinear multiobjective optimization kluwer academic publishers. *European Journal of Operational Research*, 148(1):229–230, 2003.
- [42] Elise Miller-Hooks and Hani Mahmassani. Least expected time paths in stochastic, time-varying transportation networks. *Transportation Science*, 34(2):198–215, 2000.
- [43] Yu Marco Nie and Xing Wu. Shortest path problem considering on-time arrival probability. *Transportation Research Part B: Methodological*, 43(6):597–613, 2009.
- [44] Evdokia Nikolova. Approximation algorithms for reliable stochastic combinatorial optimization. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 338–351. Springer, 2010.

- [45] Evdokia Nikolova, Jonathan A Kelner, Matthew Brand, and Michael Mitzenmacher. Stochastic shortest paths via quasi-convex maximization. In *European Symposium on Algorithms*, pages 552–563. Springer, 2006.
- [46] El-Shawarby Ihab Arafeh Mazen Rakha, Hesham and Francois Dion. Estimating path travel-time reliability. *IEEE Intelligent Transportation Systems Conference (ITSC)*, Toronto, Canada, 2006.
- [47] Mohsen Ramezani and Nikolaos Geroliminis. On the estimation of arterial route travel time distribution with markov chains. *Transportation Research Part B: Methodological*, 46(10):1576 – 1590, 2012.
- [48] Carl Rasmussen. The infinite gaussian mixture model. *Conference: Advances in Neural Information Processing Systems 12*, 11, 2000.
- [49] Adam Samara, Felix Rempe, Ulrich Fastenrath, and Simone Göttlich. Assessing the probability of arriving on time using historical travel time data in a road network. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1343–1348. IEEE, 2019.
- [50] Adam Samara, Felix Rempe, and Simone Göttlich. Modelling arterial travel time distribution using copulas. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2020.
- [51] Alexander Schickedanz, Deepak Ajwani, Ulrich Meyer, and Pawel Gawrychowski. Average-case behavior of k-shortest path algorithms. In *International Conference on Complex Networks and their Applications*, pages 28–40. Springer, 2018.
- [52] Nadine Schuessler and Kay W Axhausen. Processing raw data from global positioning systems without additional information. *Transportation Research Record*, 2105(1):28–36, 2009.
- [53] Elliott Sigal, Alan Pritsker, and James Solberg. The stochastic shortest route problem. *Operations Research*, 28(5):1122–1129, 1980.
- [54] Alex Skabardonis and Nikolaos Geroliminis. Real-time estimation of travel times along signalized arterials. In *Proceedings of the 16th International Symposium on Transportation and Traffic Theory*, 2005.
- [55] Nikolai V Smirnov. Estimate of deviation between empirical distribution functions in two independent samples. *Bulletin Moscow University*, 2(2):3–16, 1939.

- 
- [56] Terence Tao. An introduction to measure theory, *American Mathematical Society Providence*. 2011.
- [57] R Core Team. R: A language and environment for statistical computing. r foundation for statistical computing. <https://www.R-project.org/>, 2019.
- [58] Chris Tofallis. Add or multiply? a tutorial on ranking and choosing with multiple criteria. *INFORMS Transactions on education*, 14(3):109–119, 2014.
- [59] Evangelos Triantaphyllou. Multi-criteria decision making methods. In *Multi-criteria decision making methods: A comparative study*, pages 5–21. Springer, 2000.
- [60] Pravin Trivedi, David Zimmer, et al. Copula modeling: an introduction for practitioners. *Foundations and Trends® in Econometrics*, 1(1):1–111, 2007.
- [61] Hans Van Lint and Henk van Zuylen. Monitoring and predicting freeway travel time reliability: Using width and skew of day-to-day travel time distribzhouution. *Transportation Research Record*, 1917(1):54–62, 2005.
- [62] Travis Waller and Athanasios Ziliaskopoulos. On the online shortest path problem with limited arc cost dependencies. *Networks: An International Journal*, 40(4):216–227, 2002.
- [63] Yi Wang, Ning Zhang, Qixin Chen, Jingwei Yang, Chongqing Kang, and Junhui Huang. Dependent discrete convolution based probabilistic load flow for the active distribution system. *IEEE Transactions on Sustainable Energy*, 8(3):1000–1009, 2016.
- [64] Bradford Westgate, Dawn Woodard, David Matteson, and Shane Henderson. Large-network travel time distribution estimation for ambulances. *European Journal of Operational Research*, 252(1):322 – 333, 2016.
- [65] Jun Yan. Enjoy the joy of copulas: With a package copula. *Journal of Statistical Software*, 21(4):1–21, 2007.
- [66] Baiyu Yang and Elise Miller-Hooks. Adaptive routing considering delays due to signal operations. *Transportation Research Part B: Methodological*, 38(5):385–413, 2004.
- [67] Jin Y Yen. Finding the k shortest loopless paths in a network. *management Science*, 17(11):712–716, 1971.

- 
- [68] Ji Yuxiong and Michael Zhang. Travel time distributions on urban streets: Estimation with hierarchical bayesian mixture model and application to traffic analysis with high-resolution bus probe data. In *Transportation Research Board 92nd Annual Meeting*.
- [69] Milan Zeleny. Compromise programming. *Multiple criteria decision making*, 1973.
- [70] Ning Zhang, Chongqing Kang, Chanan Singh, and Qing Xia. Copula based dependent discrete convolution for power system uncertainty analysis. *IEEE Transactions on Power Systems*, 31(6):5204–5205, 2016.
- [71] Fangfang Zheng and Henk Van Zuylen. Uncertainty and predictability of urban link travel time: Delay distribution–based analysis. *Transportation Research Record*, 2192(1):136–146, 2010.